SimpleEssentials Namespace
## Classes

<table>
<thead>
<tr>
<th>Class</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Factory</td>
<td><img src="image" alt="Factory" /></td>
</tr>
</tbody>
</table>

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
Factory Class

Namespace: SimpleEssentials
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#
VB
C++

public static class Factory
Public NotInheritable Class Factory
public ref class Factory abstract sealed
Inheritance Hierarchy

Object
SimpleEssentials...Factory
See Also

Factory Members
SimpleEssentials Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
Factory Members

The Factory type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>v RegisterDefaults</td>
<td></td>
</tr>
</tbody>
</table>
### Fields

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FileHandler</td>
<td></td>
</tr>
<tr>
<td>FolderHandler</td>
<td></td>
</tr>
<tr>
<td>Log</td>
<td></td>
</tr>
</tbody>
</table>
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="Container" alt="Container" /></td>
<td>Container</td>
</tr>
</tbody>
</table>
See Also

Factory Class
SimpleEssentials Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
Factory Fields

The Factory type exposes the following members.
## Fields

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FileHandler</td>
<td></td>
</tr>
<tr>
<td>FolderHandler</td>
<td></td>
</tr>
<tr>
<td>Log</td>
<td></td>
</tr>
</tbody>
</table>
See Also

Factory Class
SimpleEssentials Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
Factory..:::FileHandler Field

Namespace:  SimpleEssentials
Assembly:  SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#
Public Shared FileHandler As FileHandler

VB
public static FileHandler FileHandler

C++
public: FileHandler^ FileHandler
See Also

- Factory Class
- SimpleEssentials Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
Factory...::...FolderHandler Field

Namespace:  SimpleEssentials
Assembly:  SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#

public static FolderHandler FolderHandler

VB

Public Shared FolderHandler As FolderHandler

C++

c Public: static FolderHandler FolderHandler

static FolderHandler FolderHandler
See Also

Factory Class
SimpleEssentials Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
Factory...Log Field

Namespace: SimpleEssentials
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#

Public Shared Log As Log

C++

public static Log Log

public:
static Log^ Log
See Also

Factory Class
SimpleEssentials Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
Factory Methods

The Factory type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>🍃 RegisterDefaults</td>
<td></td>
</tr>
</tbody>
</table>
See Also

Factory Class
SimpleEssentials Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
Factory..::..RegisterDefaults Method

**Namespace:**  SimpleEssentials  
**Assembly:**  SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#
VB
C++

public static void RegisterDefaults()

Public Shared Sub RegisterDefaults

public:
static void RegisterDefaults()
See Also

Factory Class
SimpleEssentials Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
Factory Properties

The Factory type exposes the following members.
### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>![ ]</td>
<td>Container</td>
</tr>
</tbody>
</table>
See Also

Factory Class
SimpleEssentials Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
Factory..::..Container Property

Namespace:  SimpleEssentials
Assembly:  SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#
Public Shared Readonly Property Container As Container
Get

public: static property Container^ Container {
    Container^ get ();
}

VB
Public static Container Container { get; }

C++
public static Container Container { get; }

Public Shared ReadOnly Property Container As Container
Get

public:
static property Container^ Container {
    Container^ get ();
}


See Also

Factory Class
SimpleEssentials Namespace

*Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.*
SimpleEssentials.Cache Namespace
# Classes

<table>
<thead>
<tr>
<th>Class</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CacheObject</td>
<td></td>
</tr>
<tr>
<td>CacheSettings</td>
<td></td>
</tr>
<tr>
<td>MemoryCacheManager</td>
<td></td>
</tr>
<tr>
<td>RedisCacheManager</td>
<td></td>
</tr>
<tr>
<td>SessionCacheManager</td>
<td></td>
</tr>
<tr>
<td>SessionObject&lt;Of&lt;(&lt;T&gt;)&gt;</td>
<td></td>
</tr>
</tbody>
</table>
## Interfaces

<table>
<thead>
<tr>
<th>Interface</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ICacheable</td>
<td></td>
</tr>
<tr>
<td>ICacheManager</td>
<td></td>
</tr>
<tr>
<td>ICacheObject</td>
<td></td>
</tr>
<tr>
<td>ICacheType</td>
<td></td>
</tr>
<tr>
<td>ICustomCacheObject(Of (&lt;T&gt;))</td>
<td></td>
</tr>
<tr>
<td>ISessionObject(Of (&lt;T&gt;))</td>
<td></td>
</tr>
</tbody>
</table>
## Enumerations

<table>
<thead>
<tr>
<th>Enumeration</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CacheObjectType</td>
<td></td>
</tr>
<tr>
<td>CacheStorage</td>
<td></td>
</tr>
</tbody>
</table>

*Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.*
CacheObject Class

Namespace: SimpleEssentials.Cache
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#

Public Class CacheObject
   Implements ICacheObject

VB

Public Class CacheObject
   Implements ICacheObject

C++

public class CacheObject : ICacheObject

public ref class CacheObject : ICacheObject
Inheritance Hierarchy

Object
SimpleEssentials.Cache...CacheObject
See Also

CacheObject Members
SimpleEssentials.Cache Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
CacheObject Members

The CacheObject type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Equals(System.Object)</strong></td>
<td>Determines whether the specified object is equal to the current object. (Inherited from <strong>Object</strong>.)</td>
</tr>
<tr>
<td><strong>Finalize</strong></td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from <strong>Object</strong>.)</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Serves as the default hash function. (Inherited from <strong>Object</strong>.)</td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td>Gets the <strong>Type</strong> of the current instance. (Inherited from <strong>Object</strong>.)</td>
</tr>
<tr>
<td><strong>MemberwiseClone</strong></td>
<td>Creates a shallow copy of the current <strong>Object</strong>. (Inherited from <strong>Object</strong>.)</td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td>Returns a string that represents the current object. (Inherited from <strong>Object</strong>.)</td>
</tr>
</tbody>
</table>
# Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data</td>
<td></td>
</tr>
<tr>
<td>LifeSpan</td>
<td></td>
</tr>
<tr>
<td>ObjectType</td>
<td></td>
</tr>
</tbody>
</table>
See Also

CacheObject Class
SimpleEssentials.Cache Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
CacheObject Methods

The CacheObject type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Equals(System.Object)</strong></td>
<td>Determines whether the specified object is equal to the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td><strong>Finalize</strong></td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td><strong>MemberwiseClone</strong></td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td>Returns a string that represents the current object. (Inherited from Object.)</td>
</tr>
</tbody>
</table>
See Also

CacheObject Class
SimpleEssentials.Cache Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
CacheObject Properties

The CacheObject type exposes the following members.
### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data</td>
<td></td>
</tr>
<tr>
<td>LifeSpan</td>
<td></td>
</tr>
<tr>
<td>ObjectType</td>
<td></td>
</tr>
</tbody>
</table>
See Also

CacheObject Class
SimpleEssentials.Cache Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
CacheObject...::..Data Property

Namespace:  SimpleEssentials.Cache
Assembly:  SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#

public Object Data { get; set; }

VB

Public Property Data As Object
    Get
    Set

C++

public: property Object^ Data {
    Object^ get ();
    void set (Object^ value);
}
See Also

CacheObject Class
SimpleEssentials.Cache Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
CacheObject...:.:.LifeSpan Property

Namespace: SimpleEssentials.Cache
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#

```csharp
public TimeSpan LifeSpan { get; set; }
```

VB

```vbnet
Public Property LifeSpan As TimeSpan
Get
Set
```

C++

```cpp
public: property TimeSpan^ LifeSpan {
    TimeSpan^ get ();
    void set (TimeSpan^ value);
}
```
See Also

CacheObject Class
SimpleEssentials.Cache Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
CacheObject..::..ObjectType Property

Namespace: SimpleEssentials.Cache
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#

public CacheObjectType ObjectType { get; set; }

VB

Public Property ObjectType As CacheObjectType
Get
Set

C++

public: CacheObjectType^ ObjectType { CacheObjectType^ get (); void set (CacheObjectType^ value); }

}
See Also

CacheObject Class
SimpleEssentials.Cache Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
CacheObjectType Enumeration

Namespace: SimpleEssentials.Cache
Assembly: SimpleEssentials (in SimpleEssentials.dll)
**Syntax**

C#  
VB  
C++

`public enum CacheObjectType`

`Public Enumeration CacheObjectType`

`public enum class CacheObjectType`
## Members

<table>
<thead>
<tr>
<th>Member name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>List</td>
<td></td>
</tr>
<tr>
<td>Single</td>
<td></td>
</tr>
</tbody>
</table>
See Also

SimpleEssentials.Cache Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
CacheSettings Class

Namespace: SimpleEssentials.Cache
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#
Public Class CacheSettings

VB
Public Class CacheSettings

C++
public ref class CacheSettings
Inheritance Hierarchy

Object
SimpleEssentials.Cache...CacheSettings
See Also

CacheSettings Members
SimpleEssentials.Cache Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
CacheSettings Members

The CacheSettings type exposes the following members.
## Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CacheSettings()()()</td>
<td></td>
</tr>
<tr>
<td>CacheSettings(String)</td>
<td></td>
</tr>
<tr>
<td>CacheSettings(String, TimeSpan)</td>
<td></td>
</tr>
<tr>
<td>CacheSettings(String, TimeSpan, CacheStorage)</td>
<td></td>
</tr>
<tr>
<td>CacheSettings(String, CacheStorage, TimeSpan)</td>
<td></td>
</tr>
</tbody>
</table>
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Equals(System.Object)</strong></td>
<td>Determines whether the specified object is equal to the current object. (Inherited from <strong>Object</strong>.)</td>
</tr>
<tr>
<td><strong>Finalize</strong></td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from <strong>Object</strong>.)</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Serves as the default hash function. (Inherited from <strong>Object</strong>.)</td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td>Gets the <strong>Type</strong> of the current instance. (Inherited from <strong>Object</strong>.)</td>
</tr>
<tr>
<td><strong>MemberwiseClone</strong></td>
<td>Creates a shallow copy of the current <strong>Object</strong>. (Inherited from <strong>Object</strong>.)</td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td>Returns a string that represents the current object. (Inherited from <strong>Object</strong>.)</td>
</tr>
</tbody>
</table>
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Key</td>
<td></td>
</tr>
<tr>
<td>LifeSpan</td>
<td></td>
</tr>
<tr>
<td>StorageType</td>
<td></td>
</tr>
</tbody>
</table>
See Also

CacheSettings Class
SimpleEssentials.Cache Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
CacheSettings Constructor
# Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>![CacheSettings()()()]</td>
<td></td>
</tr>
<tr>
<td>![CacheSettings(String)]</td>
<td></td>
</tr>
<tr>
<td>![CacheSettings(String, TimeSpan)]</td>
<td></td>
</tr>
<tr>
<td>![CacheSettings(String, TimeSpan, CacheStorage)]</td>
<td></td>
</tr>
<tr>
<td>![CacheSettings(String, CacheStorage, TimeSpan)]</td>
<td></td>
</tr>
</tbody>
</table>
See Also

CacheSettings Class
CacheSettings Members
SimpleEssentials.Cache Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
CacheSettings Constructor

Namespace: SimpleEssentials.Cache
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#  
VB  
C++

public CacheSettings()

Public Sub New

public:
CacheSettings()
See Also

CacheSettings Class
CacheSettings Overload
SimpleEssentials.Cache Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
CacheSettings Constructor (String)

Namespace: SimpleEssentials.Cache
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

**C#**

```csharp
public CacheSettings(
    string key
)
```

**VB**

```vbnet
Public Sub New ( _
    key As String _
)
```

**C++**

```cpp
public:
CacheSettings( 
    String^ key
)
```

**Parameters**

**key**

Type: **String**
See Also

CacheSettings Class
CacheSettings Overload
SimpleEssentials.Cache Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
CacheSettings Constructor (String, TimeSpan)

Namespace: SimpleEssentials.Cache
Assembly: SimpleEssentials (in SimpleEssentials.dll)
## Syntax

### C#

```
class CacheSettings
{
    public CacheSettings(
        string key,
        TimeSpan lifeSpan
    )
}
```

### VB

```
Public Sub New( _
    key As String, _
    lifeSpan As TimeSpan _
)
```

### C++

```
public:
CacheSettings(
    String^ key,
    TimeSpan^ lifeSpan
)
```

### Parameters

- **key**
  - Type: `String`

- **lifeSpan**
  - Type: `TimeSpan`
See Also

CacheSettings Class
CacheSettings Overload
SimpleEssentials.Cache Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
CacheSettings Constructor (String, TimeSpan, CacheStorage)

Namespace: SimpleEssentials.Cache
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

**C#**
```
public CacheSettings(
    string key,
    TimeSpan lifeSpan,
    CacheStorage storageType
)
```

**VB**
```
Public Sub New (  
    key As String,  
    lifeSpan As TimeSpan,  
    storageType As CacheStorage  
)
```

**C++**
```
public:
CacheSettings(  
    String^ key,  
    TimeSpan^ lifeSpan,  
    CacheStorage^ storageType
)
```

**Parameters**

- **key**
  Type: **String**

- **lifeSpan**
  Type: **TimeSpan**

- **storageType**
  Type: **SimpleEssentials.Cache.:::CacheStorage**
See Also

CacheSettings Class
CacheSettings Overload
SimpleEssentials.Cache Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
CacheSettings Constructor (String, CacheStorage, TimeSpan)

Namespace: SimpleEssentials.Cache
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#
VB
C++

public CacheSettings(
    string key,
    CacheStorage storageType,
    TimeSpan lifeSpan
)

Public Sub New (_
    key As String, _
    storageType As CacheStorage, _
    lifeSpan As TimeSpan _
)

public:
CacheSettings(
    String^ key,
    CacheStorage^ storageType,
    TimeSpan^ lifeSpan
)

Parameters

key
Type: String

storageType
Type: SimpleEssentials.Cache..::..CacheStorage

lifeSpan
Type: TimeSpan
See Also

CacheSettings Class
CacheSettings Overload
SimpleEssentials.Cache Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
CacheSettings Methods

The CacheSettings type exposes the following members.
# Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Equals(System.Object)</code></td>
<td>Determines whether the specified object is equal to the current object.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><code>Finalize</code></td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><code>GetHashCode</code></td>
<td>Serves as the default hash function.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><code>GetType</code></td>
<td>Gets the <code>Type</code> of the current instance.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><code>MemberwiseClone</code></td>
<td>Creates a shallow copy of the current <code>Object</code>.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><code>ToString</code></td>
<td>Returns a string that represents the current object.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <code>Object</code>.)</td>
</tr>
</tbody>
</table>
See Also

CacheSettings Class
SimpleEssentials.Cache Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
CacheSettings Properties

The CacheSettings type exposes the following members.
### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Key</td>
<td></td>
</tr>
<tr>
<td>LifeSpan</td>
<td></td>
</tr>
<tr>
<td>StorageType</td>
<td></td>
</tr>
</tbody>
</table>
See Also

CacheSettings Class
SimpleEssentials.Cache Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
CacheSettings..::..Key Property

Namespace: SimpleEssentials.Cache
Assembly: SimpleEssentials (in SimpleEssentials.dll)
**Syntax**

C#  
VB  
C++

```csharp
public string Key { get; set; }
```

```vbnet
Public Property Key As String
    Get
    Set
```

```cpp
public:
    property String^ Key {
        String^ get();
        void set(String^ value);
    }
```
See Also

CacheSettings Class
SimpleEssentials.Cache Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
CacheSettings...:...LifeSpan Property

Namespace: SimpleEssentials.Cache
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

**C#**

```csharp
public TimeSpan LifeSpan { get; set; }
```

**VB**

```vbnet
Public Property LifeSpan As TimeSpan
    Get
    Set
```

**C++**

```cpp
public: property TimeSpan^ LifeSpan { 
   (TimeSpan^ get ();
    void set (TimeSpan^ value);
}
```
See Also

CacheSettings Class
SimpleEssentials.Cache Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
CacheSettings...::StorageType Property

Namespace: SimpleEssentials.Cache
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#

public CacheStorage StorageType { get; set; }

VB

Public Property StorageType As CacheStorage
  Get
  Set

C++

public:
  property CacheStorage^ StorageType {
    CacheStorage^ get ();
    void set (CacheStorage^ value);
  }

See Also

CacheSettings Class
SimpleEssentials.Cache Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
CacheStorage Enumeration

Namespace: SimpleEssentials.Cache
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#

public enum CacheStorage

VB

Public Enumeration CacheStorage

C++

public enum class CacheStorage
### Members

<table>
<thead>
<tr>
<th>Member name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Normal</td>
<td></td>
</tr>
<tr>
<td>Hashed</td>
<td></td>
</tr>
</tbody>
</table>
See Also

SimpleEssentials.Cache Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
ICacheable Interface

Namespace:  SimpleEssentials.Cache
Assembly:  SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#
VB
C++

public interface ICacheable
Public Interface ICacheable
public interface class ICacheable
See Also

ICacheable Members
SimpleEssentials.Cache Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
ICacheable Members

The ICacheable type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals(System.Object)</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>InvalidateCache</td>
<td></td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>UpdateCache&lt;Of&lt;T&gt;&gt;</td>
<td></td>
</tr>
</tbody>
</table>
See Also

ICacheable Interface
SimpleEssentials.Cache Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
ICacheable Methods

The ICacheable type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals(System.Object)</td>
<td>Determines whether the specified object is equal to the current object.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from Object.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup</td>
</tr>
<tr>
<td></td>
<td>operations before it is reclaimed by garbage collection.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from Object.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from Object.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from Object.)</td>
</tr>
<tr>
<td>InvalidateCache</td>
<td>Creates a shallow copy of the current Object.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from Object.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Returns a string that represents the current object.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from Object.)</td>
</tr>
<tr>
<td>ToString</td>
<td></td>
</tr>
<tr>
<td>UpdateCache&lt;T&gt;</td>
<td></td>
</tr>
</tbody>
</table>
See Also

ICacheable Interface
SimpleEssentials.Cache Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
ICacheable... InvalidateCache Method

Namespace: SimpleEssentials.Cache
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#
void InvalidateCache()

VB
Sub InvalidateCache
void InvalidateCache()
See Also

ICacheable Interface
SimpleEssentials.Cache Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
ICacheable::..UpdateCache<(Of <('T')>)> Method

Namespace:  SimpleEssentials.Cache
Assembly:  SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#

void UpdateCache<T>(
    T obj
)

VB

Sub UpdateCache(Of T) ( _
    obj As T _
)

generic<typename T>
void UpdateCache(
    T obj
)

Type Parameters

T

Parameters

obj
    Type: T
See Also

ICacheable Interface
SimpleEssentials.Cache Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
ICacheManager Interface

Namespace: SimpleEssentials.Cache
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

**C#**

```csharp
public interface ICacheManager
```

**VB**

```vbnet
Public Interface ICacheManager
```

**C++**

```cpp
public interface class ICacheManager
```
See Also

ICacheManager Members
SimpleEssentials.Cache Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
ICacheManager Members

The ICacheManager type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Delete</strong></td>
<td></td>
</tr>
<tr>
<td><strong>DeleteHash</strong></td>
<td></td>
</tr>
<tr>
<td><strong>DeleteSingleHash</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Equals(System.Object)</strong></td>
<td>Determines whether the specified object is equal to the current object. (Inherited from <strong>Object</strong>.)</td>
</tr>
<tr>
<td><strong>Finalize</strong></td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from <strong>Object</strong>.)</td>
</tr>
<tr>
<td><strong>Get</strong></td>
<td></td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Serves as the default hash function. (Inherited from <strong>Object</strong>.)</td>
</tr>
<tr>
<td><strong>GetList(Of &lt;&lt;'(T)&gt;&gt;)</strong></td>
<td></td>
</tr>
<tr>
<td><strong>GetSingleHash(Of &lt;&lt;'(T)&gt;&gt;)</strong></td>
<td></td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td>Gets the <strong>Type</strong> of the current instance. (Inherited from <strong>Object</strong>.)</td>
</tr>
<tr>
<td><strong>Insert(Of &lt;&lt;'(T)&gt;&gt;)</strong></td>
<td></td>
</tr>
<tr>
<td><strong>InsertHash(Of &lt;&lt;'(T)&gt;&gt;)</strong></td>
<td></td>
</tr>
<tr>
<td><strong>InsertSingleHash(Of &lt;&lt;'(T)&gt;&gt;)</strong></td>
<td></td>
</tr>
<tr>
<td><strong>InsertSingleHash(Of &lt;&lt;'(T)&gt;&gt;)</strong>(T, CacheSettings)</td>
<td></td>
</tr>
<tr>
<td><strong>InsertSingleHash(Of &lt;&lt;'(T)&gt;&gt;)</strong>(T, CacheSettings, String)</td>
<td></td>
</tr>
</tbody>
</table>
**MemberwiseClone**  
Creates a shallow copy of the current `Object`.  
(Inherited from `Object`.)

**ToString**  
Returns a string that represents the current object.  
(Inherited from `Object`.)

**Update<(Of <<'(T)>>)  
UpdateHash<(Of <<'(T)>>)  
UpdateSingleHash<(Of <<'(T)>>)
See Also

ICacheManager Interface
SimpleEssentials.Cache Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
ICacheManager Methods

The ICacheManager type exposes the following members.
# Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Delete</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>DeleteHash</td>
<td></td>
</tr>
<tr>
<td>DeleteSingleHash</td>
<td></td>
</tr>
<tr>
<td>Equals(System.Object)</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)</td>
</tr>
<tr>
<td>Finalize</td>
<td></td>
</tr>
<tr>
<td>Get</td>
<td></td>
</tr>
<tr>
<td>GetData&lt;(Of '&lt;T&gt;&gt;&gt;()</td>
<td></td>
</tr>
<tr>
<td>GetHash&lt;(Of '&lt;T&gt;&gt;())</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td></td>
</tr>
<tr>
<td>GetList&lt;(Of '&lt;T&gt;&gt;())</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetSingleHash&lt;(Of '&lt;T&gt;&gt;())</td>
<td></td>
</tr>
<tr>
<td>GetType</td>
<td></td>
</tr>
<tr>
<td>Insert&lt;(Of '&lt;T&gt;&gt;())</td>
<td></td>
</tr>
<tr>
<td>InsertHash&lt;(Of '&lt;T&gt;&gt;())</td>
<td></td>
</tr>
<tr>
<td>InsertSingleHash&lt;(Of '&lt;T&gt;&gt;())(T, CacheSettings)</td>
<td></td>
</tr>
<tr>
<td>InsertSingleHash&lt;(Of '&lt;T&gt;&gt;())(T, CacheSettings, String)</td>
<td></td>
</tr>
</tbody>
</table>
MemberwiseClone

Creates a shallow copy of the current Object. (Inherited from Object.)

ToString

Returns a string that represents the current object. (Inherited from Object.)

Update(Of <<'(T)>>)
UpdateHash(Of <<'(T)>>)
UpdateSingleHash(Of <<'(T)>>)
See Also

ICacheManager Interface
SimpleEssentials.Cache Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
ICacheManager..::.Delete Method

Namespace: SimpleEssentials.Cache
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#

```csharp
void Delete(
    CacheSettings cacheSettings
)
```

VB

```vbnet
Sub Delete ( _
    cacheSettings As CacheSettings _
)
```

C++

```c++
void Delete(
    CacheSettings^ cacheSettings
)
```

Parameters

**cacheSettings**

Type: `SimpleEssentials.Cache::CacheSettings`
See Also

ICacheManager Interface
SimpleEssentials.Cache Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
ICacheManager..::.DeleteHash Method

Namespace: SimpleEssentials.Cache
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

**C#**

```csharp
void DeleteHash(
    CacheSettings cacheSettings
)
```

**VB**

```vb
Sub DeleteHash ( _
    cacheSettings As CacheSettings _
)
```

**C++**

```c++
void DeleteHash(
    CacheSettings^ cacheSettings
)
```

**Parameters**

cacheSettings

Type: `SimpleEssentials.Cache::<>::CacheSettings`
See Also

ICacheManager Interface
SimpleEssentials.Cache Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
ICacheManager..::..DeleteSingleHash Method

Namespace: SimpleEssentials.Cache
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#  
void DeleteSingleHash(
    CacheSettings cacheSettings,
    string fieldKey
)

VB  
Sub DeleteSingleHash ( _
    cacheSettings As CacheSettings, _
    fieldKey As String _
)

C++  
void DeleteSingleHash(
    CacheSettings cacheSettings,
    String fieldKey
)

Parameters

cacheSettings
    Type: SimpleEssentials.Cache::CacheSettings

fieldKey
    Type: String
See Also

ICacheManager Interface
SimpleEssentials.Cache Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
ICacheManager::Get Method

Namespace: SimpleEssentials.Cache
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#

```csharp
CacheObject Get(
    CacheSettings cacheSettings
)
```

Function Get ( _
    cacheSettings As CacheSettings _
) As CacheObject

```csharp
CacheObject^ Get(
    CacheSettings^ cacheSettings
)
```

VB

```vbnet
Function Get ( _
    cacheSettings As CacheSettings _
) As CacheObject

CacheObject Get(
    CacheSettings cacheSettings
)
```

C++

```cpp
CacheObject Get(
    CacheSettings cacheSettings
)
```

Parameters

cacheSettings
    Type: SimpleEssentials.Cache..::..CacheSettings
See Also

ICacheManager Interface
SimpleEssentials.Cache Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
ICacheManager..::..GetData<(Of <('T')>)> Method

Namespace: SimpleEssentials.Cache
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#
VB
C++

T GetData<T>(
    CacheSettings cacheSettings,
    string fieldKey
)

Function GetData(Of T) ( _
    cacheSettings As CacheSettings, _
    fieldKey As String _
) As T

generic<typename T>
T GetData(
    CacheSettings cacheSettings,
    String^ fieldKey
)

Type Parameters

T

Parameters

cacheSettings
    Type: SimpleEssentials.Cache::CacheSettings

fieldKey
    Type: String
See Also

ICacheManager Interface
SimpleEssentials.Cache Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
ICacheManager..::..GetHash<(Of <('T>')>) Method

Namespace: SimpleEssentials.Cache
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

```csharp
IEnumerable<T> GetHash<T>(
    CacheSettings cacheSettings
)
```

```vbnet
Function GetHash(Of T) ( _
    cacheSettings As CacheSettings _
) As IEnumerable(Of T)
```

```cpp
generic<typename T>
IEnumerable<T>^ GetHash(
    CacheSettings^ cacheSettings
)
```

Type Parameters

**T**

Parameters

**cacheSettings**

Type: **SimpleEssentials.Cache..::.CacheSettings**
See Also

ICacheManager Interface
SimpleEssentials.Cache Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
ICacheManager..::.GetList<(Of '<'T'>')>) Method

Namespace: SimpleEssentials.Cache
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#

```csharp
IEnumerable<T> GetList<T>(
    CacheSettings cacheSettings
)
```

Function GetList(0f T) ( _
    cacheSettings As CacheSettings _
) As IEnumerable(0f T)

generic<typename T>
IEnumerable<T>^ GetList(
    CacheSettings^ cacheSettings
)
```

Type Parameters

T

Parameters

cacheSettings
    Type: SimpleEssentials.Cache..::.CacheSettings
See Also

ICacheManager Interface
SimpleEssentials.Cache Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
ICacheManager..::.GetSingleHash<('T')> Method

Namespace: SimpleEssentials.Cache
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#

 VB

C++

T GetSingleHash<T>(
    CacheSettings cacheSettings,
    string fieldKey
)

Function GetSingleHash(Of T) ( _
    cacheSettings As CacheSettings, _
    fieldKey As String _
) As T

generic<typename T>
T GetSingleHash(
    CacheSettings`^ cacheSettings,
    String`^ fieldKey
)

Type Parameters

T

Parameters

cacheSettings
    Type: SimpleEssentials.Cache..::..CacheSettings

fieldKey
    Type: String
See Also

ICacheManager Interface
SimpleEssentials.Cache Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
ICacheManager..::.Insert(Of<T>)) Method

Namespace: SimpleEssentials.Cache
Assembly: SimpleEssentials (in SimpleEssentials.dll)
**Syntax**

**C#**

```csharp
void Insert<T>(
    T data,
    CacheSettings cacheSettings
)
```

**VB**

```vbnet
Sub Insert(Of T)( _
    data As T, _
    cacheSettings As CacheSettings _
)
```

**C++**

```cpp
generic<typename T>
void Insert(
    T data,
    CacheSettings^ cacheSettings
)
```

**Type Parameters**

T

**Parameters**

data

Type: T

cacheSettings

Type: SimpleEssentials.Cache:::CacheSettings
See Also

ICacheManager Interface
SimpleEssentials.Cache Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
ICacheManager..::..InsertHash<Of <(<'T'>)>> Method

Namespace: SimpleEssentials.Cache
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

**C#**

```csharp
void InsertHash<T>(
    IEnumerable<T> data,
    CacheSettings cacheSettings
)
```

**VB**

```vbnet
Sub InsertHash(Of T) (_
    data As IEnumerable(Of T), _
    cacheSettings As CacheSettings _
)
```

**C++**

```cpp
generic<typename T>
void InsertHash(
    IEnumerable<T>^ data,
    CacheSettings^ cacheSettings
)
```

**Type Parameters**

T

**Parameters**

data
    Type: IEnumerable<(Of <*>())>

cacheSettings
    Type: SimpleEssentials.Cache..::..CacheSettings
See Also

ICacheManager Interface
SimpleEssentials.Cache Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
ICacheManager::InsertSingleHash Method
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>InsertSingleHash&lt;(Of &lt;&lt;'(T)&gt;&gt;)(T, CacheSettings)</code></td>
<td></td>
</tr>
<tr>
<td><code>InsertSingleHash&lt;(Of &lt;&lt;'(T)&gt;&gt;)(T, CacheSettings, String)</code></td>
<td></td>
</tr>
</tbody>
</table>
See Also

ICacheManager Interface
ICacheManager Members
SimpleEssentials.Cache Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
ICacheManager..::..InsertSingleHash<
<('T')>> Method (T,
CacheSettings)

Namespace:  SimpleEssentials.Cache
Assembly:  SimpleEssentials (in SimpleEssentials.dll)
Syntax

**C#**

```csharp
void InsertSingleHash<T>(
    T data,
    CacheSettings cacheSettings
)
```

**VB**

```vbnet
Sub InsertSingleHash(Of T) ( _
    data As T, _
    cacheSettings As CacheSettings _
)
```

**C++**

```cpp
generic<typename T>
void InsertSingleHash(
    T data,
    CacheSettings^ cacheSettings
)
```

**Type Parameters**

**T**

**Parameters**

data
   Type: T

cacheSettings
   Type: SimpleEssentials.Cache::.CacheSettings
See Also

ICacheManager Interface
InsertSingleHash Overload
SimpleEssentials.Cache Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
ICacheManager:::..InsertSingleHash<T>() Method (T, CacheSettings, String)

Namespace: SimpleEssentials.Cache
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#

void InsertSingleHash<T>(
    T data,
    CacheSettings cacheSettings,
    string fieldKey
)

VB

Sub InsertSingleHash(Of T) ( _
    data As T, _
    cacheSettings As CacheSettings, _
    fieldKey As String _
)

c++
generic<typename T>
void InsertSingleHash(
    T data,
    CacheSettings^ cacheSettings,
    String^ fieldKey
)

Type Parameters

T

Parameters

data
    Type: T

cacheSettings
    Type: SimpleEssentials.Cache..::.CacheSettings

fieldKey
    Type: String
See Also

ICacheManager Interface
InsertSingleHash Overload
SimpleEssentials.Cache Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
ICacheManager..::.Update<(Of <'T'>)> Method

Namespace:  SimpleEssentials.Cache
Assembly:  SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#  
void Update<T>(
    T data,
    CacheSettings cacheSettings
)

VB  
Sub Update(Of T) (_
    data As T, _
    cacheSettings As CacheSettings _
)

generic<typename T>
void Update(
    T data,
    CacheSettings^ cacheSettings
)

Type Parameters

T

Parameters

data  
Type: T

cacheSettings  
Type: SimpleEssentials.Cache..::.CacheSettings
See Also

ICacheManager Interface
SimpleEssentials.Cache Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
ICacheManager..::.UpdateHash<T> Method

Namespace: SimpleEssentials.Cache
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#

```csharp
void UpdateHash<T>(
    IEnumerable<T> data,
    CacheSettings cacheSettings
)
```

VB

```vbnet
Sub UpdateHash(Of T) (_
    data As IEnumerable(Of T), _
    cacheSettings As CacheSettings _
)
```

generic<typename T>

```csharp
void UpdateHash(
    IEnumerable<T> data,
    CacheSettings cacheSettings
)
```

**Type Parameters**

T

**Parameters**

data
   Type: IEnumerable(Of (T))

cacheSettings
   Type: SimpleEssentials.Cache..::.CacheSettings
See Also

ICacheManager Interface
SimpleEssentials.Cache Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
ICacheManager::..UpdateSingleHasl
<('T>')>) Method

Namespace:  SimpleEssentials.Cache
Assembly:  SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#

```csharp
void UpdateSingleHash<T>(
    T data,
    CacheSettings cacheSettings
)
```

VB

```vb
Sub UpdateSingleHash(Of T) ( _
    data As T, _
    cacheSettings As CacheSettings _
)
```

c++

```cpp
generic<typename T>
void UpdateSingleHash(
    T data,
    CacheSettings^ cacheSettings
)
```

Type Parameters

T

Parameters

data
    Type: T

cacheSettings
    Type: SimpleEssentials.Cache::<>::CacheSettings
See Also

ICacheManager Interface
SimpleEssentials.Cache Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
ICacheObject Interface

Namespace: SimpleEssentials.Cache
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#
VB
C++

public interface ICacheObject
Public Interface ICacheObject
public interface class ICacheObject
See Also

ICacheObject Members  
SimpleEssentials.Cache Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
ICacheObject Members

The ICacheObject type exposes the following members.
### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Equals(System.Object)</strong></td>
<td>Determines whether the specified object is equal to the current object.    (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><strong>Finalize</strong></td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Serves as the default hash function. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td>Gets the Type of the current instance. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><strong>MemberwiseClone</strong></td>
<td>Creates a shallow copy of the current Object. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td>Returns a string that represents the current object. (Inherited from <code>Object</code>.)</td>
</tr>
</tbody>
</table>
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>![Data]</td>
<td>Data</td>
</tr>
<tr>
<td>![LifeSpan]</td>
<td>LifeSpan</td>
</tr>
<tr>
<td>![ObjectType]</td>
<td>ObjectType</td>
</tr>
</tbody>
</table>
See Also

ICacheObject Interface
SimpleEssentials.Cache Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
ICacheObject Methods

The ICacheObject type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Equals(System.Object)</strong></td>
<td>Determines whether the specified object is equal to the current object.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">Object</a>)</td>
</tr>
<tr>
<td><strong>Finalize</strong></td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">Object</a>)</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Serves as the default hash function.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">Object</a>)</td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td>Gets the <a href="#">Type</a> of the current instance.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">Object</a>)</td>
</tr>
<tr>
<td><strong>MemberwiseClone</strong></td>
<td>Creates a shallow copy of the current <a href="#">Object</a>.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">Object</a>)</td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td>Returns a string that represents the current object.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">Object</a>)</td>
</tr>
</tbody>
</table>
See Also

ICacheObject Interface
SimpleEssentials.Cache Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
ICacheObject Properties

The ICacheObject type exposes the following members.
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data</td>
<td></td>
</tr>
<tr>
<td>LifeSpan</td>
<td></td>
</tr>
<tr>
<td>ObjectType</td>
<td></td>
</tr>
</tbody>
</table>
See Also

ICacheObject Interface
SimpleEssentials.Cache Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
ICacheObject...::..Data Property

Namespace: SimpleEssentials.Cache
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#

VB

C++

**Object** Data { get; set; }

Property Data As **Object**
  Get
  Set

property **Object**^ Data {
  **Object**^ get ();
  void set (**Object**^ value);
}
See Also

ICacheObject Interface
SimpleEssentials.Cache Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
ICacheObject...:::LifeSpan Property

Namespace: SimpleEssentials.Cache
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#

VB

C++

```csharp
TimeSpan LifeSpan { get; set; }
```

```vbnet
Property LifeSpan As TimeSpan
    Get
    Set

property TimeSpan^ LifeSpan {
    TimeSpan^ get () ;
    void set (TimeSpan^ value );
}
```
See Also

ICacheObject Interface
SimpleEssentials.Cache Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
ICacheObject...::ObjectType

Namespace: SimpleEssentials.Cache
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#

VB

C++

```
CacheObjectType ObjectType { get; set; }

Property ObjectType As CacheObjectType
    Get
    Set

property CacheObjectType^ ObjectType {
    CacheObjectType^ get ();
    void set (CacheObjectType^ value);
}
```
See Also

ICacheObject Interface
SimpleEssentials.Cache Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
ICacheType Interface

Namespace: SimpleEssentials.Cache
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#  
public interface ICacheType

VB  
Public Interface ICacheType

C++  
public interface class ICacheType
See Also

ICacheType Members
SimpleEssentials.Cache Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
ICacheType Members

The ICacheType type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Delete</strong></td>
<td></td>
</tr>
<tr>
<td><strong>DeleteHash</strong></td>
<td></td>
</tr>
<tr>
<td><strong>DeleteSingleHash</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Equals(System.Object)</strong></td>
<td>Determines whether the specified object is equal to the current object.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <strong>Object</strong>.)</td>
</tr>
<tr>
<td><strong>Finalize</strong></td>
<td>Allows an object to try to free resources and perform other cleanup</td>
</tr>
<tr>
<td></td>
<td>operations before it is reclaimed by garbage collection.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <strong>Object</strong>.)</td>
</tr>
<tr>
<td><strong>Get&lt;(Of &lt;&lt;'(T))&gt;&gt;</strong></td>
<td></td>
</tr>
<tr>
<td><strong>GetHash&lt;(Of &lt;&lt;'(T))&gt;&gt;</strong></td>
<td></td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Serves as the default hash function.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <strong>Object</strong>.)</td>
</tr>
<tr>
<td><strong>GetList&lt;(Of &lt;&lt;'(T)&gt;&gt;)&gt;</strong></td>
<td></td>
</tr>
<tr>
<td><strong>GetSingleHash&lt;(Of &lt;&lt;'(T)&gt;&gt;)&gt;</strong></td>
<td></td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td>Gets the <strong>Type</strong> of the current instance.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <strong>Object</strong>.)</td>
</tr>
<tr>
<td><strong>Insert&lt;(Of &lt;&lt;'(T)&gt;&gt;)&gt;&gt;</strong></td>
<td></td>
</tr>
<tr>
<td><strong>InsertHash&lt;(Of &lt;&lt;'(T)&gt;&gt;)&gt;&gt;</strong></td>
<td></td>
</tr>
<tr>
<td><strong>InsertSingleHash&lt;(Of &lt;&lt;'(T)&gt;&gt;)&gt;&gt;</strong></td>
<td>Creates a shallow copy of the current <strong>Object</strong>.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <strong>Object</strong>.)</td>
</tr>
<tr>
<td><strong>MemberwiseClone</strong></td>
<td></td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td>Returns a string that represents the current object.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <strong>Object</strong>.)</td>
</tr>
</tbody>
</table>
Update(Of <<'(T)>>)
UpdateHash(Of <<'(T)>>)
UpdateSingleHash(Of <<'(T)>>)
See Also

ICacheType Interface
SimpleEssentials.Cache Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
ICacheType Methods

The ICacheType type exposes the following members.
# Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Delete</strong></td>
<td></td>
</tr>
<tr>
<td><strong>DeleteHash</strong></td>
<td></td>
</tr>
<tr>
<td><strong>DeleteSingleHash</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Equals(System.Object)</strong></td>
<td>Determines whether the specified object is equal to the current object. (Inherited from <strong>Object</strong>.)</td>
</tr>
<tr>
<td><strong>Finalize</strong></td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from <strong>Object</strong>.)</td>
</tr>
<tr>
<td><strong>Get&lt;(Of &lt;&lt;'(T)&gt;&gt;&gt;)</strong></td>
<td></td>
</tr>
<tr>
<td><strong>GetHash&lt;(Of &lt;&lt;'(T)&gt;&gt;&gt;)</strong></td>
<td></td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Serves as the default hash function. (Inherited from <strong>Object</strong>.)</td>
</tr>
<tr>
<td><strong>GetList&lt;(Of &lt;&lt;'(T)&gt;&gt;&gt;)</strong></td>
<td></td>
</tr>
<tr>
<td><strong>GetSingleHash&lt;(Of &lt;&lt;'(T)&gt;&gt;&gt;)</strong></td>
<td></td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td>Gets the <strong>Type</strong> of the current instance. (Inherited from <strong>Object</strong>.)</td>
</tr>
<tr>
<td><strong>Insert&lt;(Of &lt;&lt;'(T)&gt;&gt;&gt;)</strong></td>
<td></td>
</tr>
<tr>
<td><strong>InsertHash&lt;(Of &lt;&lt;'(T)&gt;&gt;&gt;)</strong></td>
<td></td>
</tr>
<tr>
<td><strong>InsertSingleHash&lt;(Of &lt;&lt;'(T)&gt;&gt;&gt;)</strong></td>
<td></td>
</tr>
<tr>
<td><strong>MemberwiseClone</strong></td>
<td>Creates a shallow copy of the current <strong>Object</strong>. (Inherited from <strong>Object</strong>.)</td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td>Returns a string that represents the current object. (Inherited from <strong>Object</strong>.)</td>
</tr>
</tbody>
</table>
Update<(O presses
"'T')></>
UpdateHash<(O presses
"'T')></>
UpdateSingleHash<(O presses
"'T')></>
See Also

ICacheType Interface
SimpleEssentials.Cache Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
ICacheType...::..Delete Method

Namespace: SimpleEssentials.Cache
Assembly: SimpleEssentials (in SimpleEssentials.dll)
**Syntax**

C#  
```csharp
void Delete(
    string cacheKey
)
```

VB  
```vbnet
Sub Delete (_
    cacheKey As String _
)
```

C++  
```cpp
void Delete(
    String cacheKey
)
```

**Parameters**

cacheKey  
Type: `String`
See Also

ICacheType Interface
SimpleEssentials.Cache Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
ICacheType...::..DeleteHash Method

Namespace: SimpleEssentials.Cache
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#  
void DeleteHash(
    string cacheKey
)

Sub DeleteHash ( _
    cacheKey As String _
)

void DeleteHash(
    String^ cacheKey
)

Parameters

cacheKey
    Type: String
See Also

ICacheType Interface
SimpleEssentials.Cache Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
ICacheType...::..DeleteSingleHash Method

Namespace: SimpleEssentials.Cache
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

**C#**

```csharp
void DeleteSingleHash(
    string cacheKey,
    string fieldKey
)
```

**VB**

```vbnet
Sub DeleteSingleHash ( _
    cacheKey As String, _
    fieldKey As String _
)
```

```vbnet
void DeleteSingleHash(
    String^ cacheKey,
    String^ fieldKey
)
```

**Parameters**

- **cacheKey**
  - Type: **String**

- **fieldKey**
  - Type: **String**
See Also

ICacheType Interface
SimpleEssentials.Cache Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
ICacheType...::.Get(Of (<'T>))> Method

Namespace: SimpleEssentials.Cache
Assembly: SimpleEssentials (in SimpleEssentials.dll)
### Syntax

**C#**

```csharp
T Get<T>(
    string cacheKey
)
```

**VB**

```vbnet
Function Get(Of T) ( _
    cacheKey As String _
) As T
```

generic<typename T>

```csharp
T Get(?
    String^ cacheKey
)
```

### Type Parameters

T

### Parameters

cacheKey

  Type: String
See Also

ICacheType Interface
SimpleEssentials.Cache Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
ICacheType::GetHash<Of <('T')>> Method

Namespace: SimpleEssentials.Cache
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#

```csharp
IEnumerable<T> GetHash<T>(
    string cacheKey
)
```

VB

```vbnet
Function GetHash(Of T) ( _
    cacheKey As String _
) As IEnumerable(Of T)
```

generic<typename T>

```csharp
IEnumerable<T>^ GetHash(
    String^ cacheKey
)
```

Type Parameters

T

Parameters

cacheKey
Type: String
See Also

ICacheType Interface
SimpleEssentials.Cache Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
ICacheType...::..GetList<(Of
<('T')>)> Method

Namespace: SimpleEssentials.Cache
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#  
VB  
C++

```csharp
IEnumerable<T> GetList<T>(
    string cacheKey
)

Function GetList(Of T) ( _
    cacheKey As String _
) As IEnumerable(Of T)

generic<typename T>
IEnumerable<T>^ GetList(
    String^ cacheKey
)
```

Type Parameters

T

Parameters

cacheKey
    Type: String
See Also

ICacheType Interface
SimpleEssentials.Cache Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
ICacheType...::..GetSingleHash<(Of <('T')>)> Method

Namespace: SimpleEssentials.Cache
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#  
```csharp
T GetSingleHash<T>(
    string cacheKey,
    string fieldKey
)
```

VB  
```vbnet
Function GetSingleHash(Of T) ( _
    cacheKey As String, _
    fieldKey As String _
) As T
```

C++
```cpp
generic<typename T>
T GetSingleHash(
    String^ cacheKey,
    String^ fieldKey
)
```

Type Parameters

T

Parameters

cacheKey  
Type: String

fieldKey  
Type: String
See Also

ICacheType Interface
SimpleEssentials.Cache Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
ICacheType...::..Insert<(Of <(<'T'>)>)> Method

Namespace: SimpleEssentials.Cache
Assembly: SimpleEssentials (in SimpleEssentials.dll)
## Syntax

**C#**

```csharp
void Insert<T>(
    T data,
    string cacheKey,
    Nullable<TimeSpan> lifeSpan
)
```

**VB**

```vbnet
Sub Insert(Of T) ( _
    data As T, _
    cacheKey As String, _
    lifeSpan As Nullable(Of TimeSpan) _
)
```

**C++**

```cpp
generic<typename T>
void Insert(
    T data,
    String^ cacheKey,
    Nullable<TimeSpan^> lifeSpan
)
```

### Type Parameters

- **T**

### Parameters

- **data**
  - Type: `T`

- **cacheKey**
  - Type: `String`

- **lifeSpan**
  - Type: `Nullable<TimeSpan>,null`
See Also

ICacheType Interface
SimpleEssentials.Cache Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
ICacheType..::..InsertHash<(Of <("T">))> Method

Namespace:  SimpleEssentials.Cache  
Assembly:  SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#

void InsertHash<T>(
    IEnumerable<T> data,
    string cacheKey,
    Nullable<TimeSpan> lifeSpan
)

Sub InsertHash(Of T) (_
    data As IEnumerable(Of T), _
    cacheKey As String, _
    lifeSpan As Nullable(Of TimeSpan) _
)

generic<typename T>
void InsertHash(
    IEnumerable<T>^ data,
    String^ cacheKey,
    Nullable<TimeSpan^> lifeSpan
)

Type Parameters

T

Parameters

data
    Type: IEnumerable(Of (Of<T>))>

cacheKey
    Type: String

lifeSpan
    Type: Nullable(Of (Of TimeSpan)>)}
See Also

ICacheType Interface
SimpleEssentials.Cache Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
ICacheType..::.InsertSingleHash<Of <('T')?>> Method

Namespace: SimpleEssentials.Cache
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#
VB
C++

void InsertSingleHash<T>(
    T data,
    string cacheKey,
    Nullable<TimeSpan> lifeSpan
)

Sub InsertSingleHash(Of T) (_
    data As T, _
    cacheKey As String, _
    lifeSpan As Nullable(Of TimeSpan) _
)

generic<typename T>
void InsertSingleHash(
    T data,
    String^ cacheKey,
    Nullable<TimeSpan^> lifeSpan
)

Type Parameters

T

Parameters

data
    Type: T

cacheKey
    Type: String

lifeSpan
    Type: Nullable<((Of (Nullable<TimeSpan>)>)>
See Also

ICacheType Interface
SimpleEssentials.Cache Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
ICacheType...::..Update<Of <('T')>> Method

Namespace: SimpleEssentials.Cache
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#  
VB  
C++

void Update<T>(
    T data,
    string cacheKey,
    Nullable<TimeSpan> lifeSpan
)

Sub Update(Of T) ( _
    data As T, _
    cacheKey As String, _
    lifeSpan As Nullable(Of TimeSpan) _
)

generic<typename T>
void Update(
    T data,
    String^ cacheKey,
    Nullable<TimeSpan^> lifeSpan
)

Type Parameters

T

Parameters

data  
    Type: T

(cacheKey  
    Type: String

lifeSpan  
    Type: Nullable(Of (Of <TimeSpan>)>)
See Also

ICacheType Interface
SimpleEssentials.Cache Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
ICacheType..::..UpdateHash<(Of <('T')>)> Method

Namespace: SimpleEssentials.Cache
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#

```csharp
void UpdateHash<T>(
    IEnumerable<T> data,
    string cacheKey,
    Nullable<TimeSpan> lifeSpan
)
```

VB

```vb
Sub UpdateHash(Of T) ( _
    data As IEnumerable(Of T), _
    cacheKey As String, _
    lifeSpan As Nullable(Of TimeSpan) _
)
```

c++

```cpp
generic<typename T>
void UpdateHash(
    IEnumerable<T> data,
    String cacheKey,
    Nullable<TimeSpan> lifeSpan
)
```

Type Parameters

T

Parameters

data
    Type: IEnumerable(Of (T))

(cacheKey
    Type: String

lifeSpan
    Type: Nullable(Of TimeSpan)
See Also

ICacheType Interface
SimpleEssentials.Cache Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
ICacheType...::..UpdateSingleHash<(C
<('T'>))> Method

Namespace: SimpleEssentials.Cache
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#

void UpdateSingleHash<T>(
    T data,
    string cacheKey,
    Nullable<TimeSpan> lifeSpan
)

VB

Sub UpdateSingleHash(Of T) ( _
    data As T, _
    cacheKey As String, _
    lifeSpan As Nullable(Of TimeSpan) _
)

cpp

generic<typename T>
void UpdateSingleHash(
    T data,
    String^ cacheKey,
    Nullable<TimeSpan^> lifeSpan
)

Type Parameters

T

Parameters

data
    Type: T

cacheKey
    Type: String

lifeSpan
    Type: Nullable<(Of <('TimeSpan')>)>
See Also

ICacheType Interface
SimpleEssentials.Cache Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
ICustomCacheObject<(Of <('T')>)> Interface

Namespace: SimpleEssentials.Cache
Assembly: SimpleEssentials (in SimpleEssentials.dll)
**Syntax**

**C#**
```csharp
public interface ICustomCacheObject<T>
```

**VB**
```vbnet
Public Interface ICustomCacheObject(Of T)
```

**C++**
```cpp
generic<typename T>
policy interface class ICustomCacheObject
```

**Type Parameters**

T
See Also

ICustomCacheObject Members
SimpleEssentials.Cache Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
ICustomCacheObject Members

The ICustomCacheObject<((Of (<T>))>) type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Equals(System.Object)</code></td>
<td>Determines whether the specified object is equal to the current object.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><code>Finalize</code></td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><code>GetHashCode</code></td>
<td>Serves as the default hash function.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><code>GetType</code></td>
<td>Gets the <code>Type</code> of the current instance.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><code>MemberwiseClone</code></td>
<td>Creates a shallow copy of the current <code>Object</code>.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><code>ToString</code></td>
<td>Returns a string that represents the current object.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
</tr>
<tr>
<td>Data</td>
<td></td>
</tr>
<tr>
<td>LifeSpan</td>
<td></td>
</tr>
</tbody>
</table>
See Also

ICustomCacheObject<(Of <('T')>)> Interface
SimpleEssentials.Cache Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
ICustomCacheObject Methods

The `ICustomCacheObject<Of `<T>`>` type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="System.Object" alt="Equals" /></td>
<td>Determines whether the specified object is equal to the current object.</td>
</tr>
<tr>
<td><img src="Object" alt="Finalize" /></td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection.</td>
</tr>
<tr>
<td><img src="Object" alt="GetHashCode" /></td>
<td>Serves as the default hash function.</td>
</tr>
<tr>
<td><img src="Object" alt="GetType" /></td>
<td>Gets the <strong>Type</strong> of the current instance.</td>
</tr>
<tr>
<td><img src="Object" alt="MemberwiseClone" /></td>
<td>Creates a shallow copy of the current <strong>Object</strong>.</td>
</tr>
<tr>
<td><img src="Object" alt="ToString" /></td>
<td>Returns a string that represents the current object.</td>
</tr>
</tbody>
</table>
See Also

ICustomCacheObject(Of 'T) Interface
SimpleEssentials.Cache Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
ICustomCacheObject Properties

The ICustomCacheObject<Of <(T)>> type exposes the following members.
### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data</td>
<td></td>
</tr>
<tr>
<td>LifeSpan</td>
<td></td>
</tr>
</tbody>
</table>
See Also

ICustomCacheObject(Of (Of 'T)>)> Interface
SimpleEssentials.Cache Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
ICustomCacheObject<

('T')>

......Data Property

Namespace: SimpleEssentials.Cache
Assembly: SimpleEssentials (in SimpleEssentials.dll)
**Syntax**

**C#**

```csharp
T Data { get; set; }
```

**VB**

```vbnet
Property Data As T
    Get
    Set
```

**C++**

```cpp
property T Data {
    T get ();
    void set (T value);
}
```
See Also

ICustomCacheObject<(Of <(T)>)> Interface
SimpleEssentials.Cache Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
ICustomCacheObject<(Of )>(('T')>)...:..:LifeSpan Property

Namespace: SimpleEssentials.Cache
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#

VB

C++

_TimeSpan_ LifeSpan { get; set; }

Property LifeSpan As _TimeSpan_
Get
Set

property _TimeSpan^_ LifeSpan {
_TimeSpan^_ get ();
void set (_TimeSpan^_ value);
}


See Also

ICustomCacheObject(Of T) Interface
SimpleEssentials.Cache Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
ISessionObject<T> Interface

Namespace: SimpleEssentials.Cache
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

**C#**

```csharp
public interface ISessionObject<T>

Public Interface ISessionObject(Of T)

generic<typename T>
public interface class ISessionObject
```

**Type Parameters**

T
See Also

ISessionObject Members
SimpleEssentials.Cache Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
**ISessionObject Members**

The **ISessionObject**<**(Of (<>'))]** type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Equals(System.Object)</code></td>
<td>Determines whether the specified object is equal to the current object.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">Object</a>)</td>
</tr>
<tr>
<td><code>Finalize</code></td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">Object</a>)</td>
</tr>
<tr>
<td><code>GetHashCode</code></td>
<td>Serves as the default hash function.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">Object</a>)</td>
</tr>
<tr>
<td><code>GetType</code></td>
<td>Gets the <a href="#">Type</a> of the current instance.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">Object</a>)</td>
</tr>
<tr>
<td><code>MemberwiseClone</code></td>
<td>Creates a shallow copy of the current <a href="#">Object</a>.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">Object</a>)</td>
</tr>
<tr>
<td><code>ToString</code></td>
<td>Returns a string that represents the current object.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">Object</a>)</td>
</tr>
</tbody>
</table>
# Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data</td>
<td></td>
</tr>
<tr>
<td>Expiration</td>
<td></td>
</tr>
</tbody>
</table>
See Also

**ISessionObject(Of T) Interface**

**SimpleEssentials.Cache Namespace**

*Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.*
ISessionObject Methods

The ISessionObject<Of <('T)>)> type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Equals(System.Object)</strong></td>
<td>Determines whether the specified object is equal to the current object. (Inherited from <strong>Object</strong>.)</td>
</tr>
<tr>
<td><strong>Finalize</strong></td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from <strong>Object</strong>.)</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Serves as the default hash function. (Inherited from <strong>Object</strong>.)</td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td>Gets the <strong>Type</strong> of the current instance. (Inherited from <strong>Object</strong>.)</td>
</tr>
<tr>
<td><strong>MemberwiseClone</strong></td>
<td>Creates a shallow copy of the current <strong>Object</strong>. (Inherited from <strong>Object</strong>.)</td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td>Returns a string that represents the current object. (Inherited from <strong>Object</strong>.)</td>
</tr>
</tbody>
</table>
See Also

I$essionObject<$Of <$T$)>$ Interface
SimpleEssentials.Cache Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
The `ISessionObject<Of <('T')>>` type exposes the following members.
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data</td>
<td></td>
</tr>
<tr>
<td>Expiration</td>
<td></td>
</tr>
</tbody>
</table>
See Also

IบางคนObject<(Of <(<'T'>)>)> Interface
SimpleEssentials.Cache Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
ISessionObject<(Of <('T')>)>...:..Data Property

Namespace: SimpleEssentials.Cache
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#  
VB  
C++

T Data { get; set; }

Property Data As T
    Get
    Set

property T Data {
    T get ();
    void set (T value);
}
See Also

ISessionObject<(Of <(<T>)>)> Interface
SimpleEssentials.Cache Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
ISessionObject<
<
<T>
>
>::Expiration Property

Namespace: SimpleEssentials.Cache
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#

VB

C++

Nullable<DateTime> Expiration { get; set; }

Property Expiration As Nullable(Of DateTime)
Get
Set

property Nullable<DateTime>^ Expiration {
Nullable<DateTime>^ get ();
void set (Nullable<DateTime>^ value);
}


See Also

I SESSION OBJECT<(OF <(T)>)> INTERFACE
SimpleEssentials.Cache Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
MemoryCacheManager Class

Namespace: SimpleEssentials.Cache
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#
VB
C++

public class MemoryCacheManager : ICacheManager

Public Class MemoryCacheManager _
    Implements ICacheManager

public ref class MemoryCacheManager : ICacheManager
Inheritance Hierarchy

Object
SimpleEssentials.Cache...MemoryCacheManager
See Also

MemoryCacheManager Members
SimpleEssentials.Cache Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
MemoryCacheManager Members

The MemoryCacheManager type exposes the following members.
## Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MemoryCacheManager</td>
<td></td>
</tr>
</tbody>
</table>
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Delete</strong></td>
<td></td>
</tr>
<tr>
<td><strong>DeleteHash</strong></td>
<td></td>
</tr>
<tr>
<td><strong>DeleteSingleHash</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Equals(System.Object)</strong></td>
<td>Determines whether the specified object is equal to the current object.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <strong>Object</strong>.)</td>
</tr>
<tr>
<td><strong>Finalize</strong></td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <strong>Object</strong>.)</td>
</tr>
<tr>
<td><strong>Get</strong></td>
<td></td>
</tr>
<tr>
<td><strong>GetDataType&lt;(Of &lt;&lt;'(T)&gt;&gt;)&gt;</strong></td>
<td></td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Serves as the default hash function.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <strong>Object</strong>.)</td>
</tr>
<tr>
<td><strong>GetList&lt;(Of &lt;&lt;'(T)&gt;&gt;)&gt;</strong></td>
<td></td>
</tr>
<tr>
<td><strong>GetSingleHash&lt;(Of &lt;&lt;'(T)&gt;&gt;)&gt;</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td>Gets the <strong>Type</strong> of the current instance.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <strong>Object</strong>.)</td>
</tr>
<tr>
<td><strong>Insert&lt;(Of &lt;&lt;'(T)&gt;&gt;)&gt;</strong></td>
<td></td>
</tr>
<tr>
<td><strong>InsertHash&lt;(Of &lt;&lt;'(T)&gt;&gt;)&gt;</strong></td>
<td></td>
</tr>
<tr>
<td><strong>InsertSingleHash&lt;(Of &lt;&lt;'(T)&gt;&gt;)&gt;</strong></td>
<td></td>
</tr>
<tr>
<td><strong>InsertSingleHash&lt;(Of &lt;&lt;'(T)&gt;&gt;)&gt;</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
- **MemberwiseClone**: Creates a shallow copy of the current `Object`. (Inherited from `Object`.)
- **ToString**: Returns a string that represents the current object. (Inherited from `Object`.)
- **Update(Of <<'(T)>>()**
- **UpdateHash(Of <<'(T)>>()**
- **UpdateSingleHash(Of <<'(T)>>()**
See Also

MemoryCacheManager Class
SimpleEssentials.Cache Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
MemoryCacheManager Constructor

Namespace: SimpleEssentials.Cache
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#

VB

C++

public MemoryCacheManager()

Public Sub New

public:
MemoryCacheManager()
See Also

MemoryCacheManager Class
SimpleEssentials.Cache Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
MemoryCacheManager Methods

The MemoryCacheManager type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Delete" /></td>
<td>Determines whether the specified object is equal to the current object.</td>
</tr>
<tr>
<td><img src="image" alt="DeleteHash" /></td>
<td>(Inherited from Object.)</td>
</tr>
<tr>
<td><img src="image" alt="DeleteSingleHash" /></td>
<td></td>
</tr>
<tr>
<td>![Equals(System.Object)]</td>
<td></td>
</tr>
<tr>
<td><img src="image" alt="Finalize" /></td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection.</td>
</tr>
<tr>
<td><img src="image" alt="Get" /></td>
<td>(Inherited from Object.)</td>
</tr>
<tr>
<td>![GetData(Of &lt;&lt;'(T))&gt;&gt;)]</td>
<td></td>
</tr>
<tr>
<td>![GetHash(Of &lt;&lt;'(T)&gt;&gt;)]</td>
<td></td>
</tr>
<tr>
<td><img src="image" alt="GetHashCode" /></td>
<td>Serves as the default hash function.</td>
</tr>
<tr>
<td>![GetList(Of &lt;&lt;'(T)&gt;&gt;)]</td>
<td>(Inherited from Object.)</td>
</tr>
<tr>
<td>![GetSingleHash(Of &lt;&lt;'(T)&gt;&gt;)]</td>
<td></td>
</tr>
<tr>
<td><img src="image" alt="GetType" /></td>
<td>Gets the Type of the current instance.</td>
</tr>
<tr>
<td>![Insert(Of &lt;&lt;'(T)&gt;&gt;)]</td>
<td>(Inherited from Object.)</td>
</tr>
<tr>
<td>![InsertHash(Of &lt;&lt;'(T)&gt;&gt;)]</td>
<td></td>
</tr>
<tr>
<td>![InsertSingleHash(Of &lt;&lt;'(T)&gt;&gt;)](T, CacheSettings)</td>
<td></td>
</tr>
<tr>
<td>![InsertSingleHash(Of &lt;&lt;'(T)&gt;&gt;)](T, CacheSettings, String)</td>
<td></td>
</tr>
</tbody>
</table>
MemberwiseClone

Creates a shallow copy of the current Object.
(Inherited from Object.)

ToString

Returs a string that represents the current object.
(Inherited from Object.)

Update(Of <<(T)>>)

UpdateHash(Of <<(T)>>)

UpdateSingleHash(Of <<(T)>>)
See Also

MemoryCacheManager Class
SimpleEssentials.Cache Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
MemoryCacheManager..::.Delete Method

Namespace: SimpleEssentials.Cache
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#

```csharp
public void Delete(
    CacheSettings cacheSettings
)
```

VB

```vbnet
Public Sub Delete (_
    cacheSettings As CacheSettings _
)
```

C++

```cpp
public:
void Delete(
    CacheSettings^ cacheSettings
)
```

**Parameters**

cacheSettings

Type: SimpleEssentials.Cache::::CacheSettings
See Also

MemoryCacheManager Class
SimpleEssentials.Cache Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
MemoryCacheManager::DeleteHas Method

Namespace: SimpleEssentials.Cache
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#

```csharp
public void DeleteHash(
    CacheSettings cacheSettings
)
```

VB

```vbnet
Public Sub DeleteHash (_
    cacheSettings As CacheSettings _
)
```

C++

```cpp
public:
void DeleteHash(
    CacheSettings^ cacheSettings
)
```

Parameters

cacheSettings
  Type: SimpleEssentials.Cache::CacheSettings
See Also

MemoryCacheManager Class
SimpleEssentials.Cache Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
MemoryCacheManager..::.DeleteSingleHash Method

Namespace: SimpleEssentials.Cache
Assembly: SimpleEssentials (in SimpleEssentials.dll)
## Syntax

### C#

```csharp
public void DeleteSingleHash(
    CacheSettings cacheSettings,
    string fieldKey
)
```

### VB

```vbnet
Public Sub DeleteSingleHash ( _
    cacheSettings As CacheSettings, _
    fieldKey As String _
)
```

### C++

```cpp
public:
void DeleteSingleHash(
    CacheSettings^ cacheSettings,
    String^ fieldKey
)
```

## Parameters

- **cacheSettings**
  - Type: `SimpleEssentials.Cache::CacheSettings`

- **fieldKey**
  - Type: `String`
See Also

MemoryCacheManager Class
SimpleEssentials.Cache Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
MemoryCacheManager...Get Method

Namespace: SimpleEssentials.Cache
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#

```csharp
public CacheObject Get(
    CacheSettings cacheSettings
)
```

VB

```vbnet
Public Function Get ( _
    cacheSettings As CacheSettings _
) As CacheObject
```

C++

```cpp
public: 
CacheObject ^ Get(
    CacheSettings ^ cacheSettings
)
```

Parameters

cacheSettings
   Type: SimpleEssentials.Cache..::..CacheSettings
See Also

MemoryCacheManager Class
SimpleEssentials.Cache Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
MemoryCacheManager...GetData<(<'T'>)> Method

Namespace: SimpleEssentials.Cache
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#  
public T GetData<T>(
    CacheSettings cacheSettings,
    string fieldKey
)

VB  
Public Function GetData(Of T) ( _
    cacheSettings As CacheSettings, _
    fieldKey As String _
) As T

C++
public:
generic<typename T>
T GetData(
    CacheSettings^ cacheSettings,
    String^ fieldKey
)

Type Parameters

T

Parameters

cacheSettings  
Type: SimpleEssentials.Cache:::CacheSettings

fieldKey  
Type: String
See Also

MemoryCacheManager Class
SimpleEssentials.Cache Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
MemoryCacheManager...:::GetHash<
<('T')>> Method

Namespace: SimpleEssentials.Cache
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#

```csharp
public IEnumerable<T> GetHash<T>(
    CacheSettings cacheSettings)
```

VB

```vbnet
Public Function GetHash(Of T) ( _
    cacheSettings As CacheSettings _
) As IEnumerable(Of T)
```

C++

```cpp
public:
    template<typename T>
    IEnumerable<T>^ GetHash(
        CacheSettings^ cacheSettings
    )
```

Type Parameters

T

Parameters

cacheSettings
    Type: SimpleEssentials.Cache...CacheSettings
See Also

MemoryCacheManager Class
SimpleEssentials.Cache Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
MemoryCacheManager..::..GetList<'( (<'T'>)')> Method

Namespace: SimpleEssentials.Cache
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#

```csharp
public IEnumerable<T> GetList<T>(
    CacheSettings cacheSettings
)
```

VB

```vbnet
Public Function GetList(Of T) (_, cacheSettings As CacheSettings _) As IEnumerable(Of T)
```

C++

```cpp
public:

generic<typename T>

IEnumerable<T>^ GetList(
    CacheSettings^ cacheSettings
)
```

Type Parameters

T

Parameters

cacheSettings

Type: SimpleEssentials.Cache::<>::CacheSettings
See Also

MemoryCacheManager Class
SimpleEssentials.Cache Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
MemoryCacheManager..::.GetSingle<('T')> Method

Namespace: SimpleEssentials.Cache
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#

public T GetSingleHash<T>(
            CacheSettings cacheSettings,
            string fieldKey
)

VB

Public Function GetSingleHash(Of T) ( _
            cacheSettings As CacheSettings, _
            fieldKey As String _
) As T

C++

public:
    template<typename T>
    T GetSingleHash(
            CacheSettings^ cacheSettings,
            String^ fieldKey
    )

Type Parameters

T

Parameters

- cacheSettings
  Type: SimpleEssentials.Cache::..::CacheSettings

- fieldKey
  Type: String
See Also

MemoryCacheManager Class
SimpleEssentials.Cache Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
MemoryCacheManager..::..Insert< Of <('T')>> Method

Namespace: SimpleEssentials.Cache
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#
Public Sub Insert(Of T) ( _
    data As T, _
    cacheSettings As CacheSettings _
)

class CacheSettings

Type Parameters

T

Parameters

data
    Type: T

cacheSettings
    Type: SimpleEssentials.Cache..::.CacheSettings
See Also

MemoryCacheManager Class
SimpleEssentials.Cache Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
MemoryCacheManager...:::InsertHasl<('T')>::> Method

Namespace: SimpleEssentials.Cache
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#

```csharp
public void InsertHash<T>(
    IEnumerable<T> data,
    CacheSettings cacheSettings
)
```

VB

```vbnet
Public Sub InsertHash(Of T) (_
    data As IEnumerable(Of T), _
    cacheSettings As CacheSettings _) =
```

C++

```c++
public:
    template<typename T>
    void InsertHash(
        IEnumerable<T> data,
        CacheSettings cacheSettings
    )
```

Type Parameters

**T**

Parameters

data
  Type: **IEnumerable<(Of (<'T'>))>

cacheSettings
  Type: **SimpleEssentials.Cache..::.CacheSettings**
See Also

MemoryCacheManager Class
SimpleEssentials.Cache Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
MemoryCacheManager::InsertSingleHash Method
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>InsertSingleHash&lt;(Of &lt;&lt;(T)&gt;&gt;)(T, CacheSettings)</td>
<td></td>
</tr>
<tr>
<td>InsertSingleHash&lt;(Of &lt;&lt;(T)&gt;&gt;)(T, CacheSettings, String)</td>
<td></td>
</tr>
</tbody>
</table>
See Also

MemoryCacheManager Class
MemoryCacheManager Members
SimpleEssentials.Cache Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
MemoryCacheManager...InsertSing<T> Method (T, CacheSettings)

Namespace: SimpleEssentials.Cache
Assembly: SimpleEssentials (in SimpleEssentials.dll)
# Syntax

**C#**

```csharp
public void InsertSingleHash<T>(
    T data,
    CacheSettings cacheSettings
)
```

**VB**

```vbnet
Public Sub InsertSingleHash(Of T) ( _
    data As T, _
    cacheSettings As CacheSettings _
)
```

**C++**

```c++
public:
    generic<typename T>
    void InsertSingleHash( 
        T data,
        CacheSettings^ cacheSettings 
    )
```

## Type Parameters

**T**

## Parameters

**data**

Type: **T**

**cacheSettings**

Type: [SimpleEssentials.Cache...CacheSettings](#)
See Also

MemoryCacheManager Class
InsertSingleHash Overload
SimpleEssentials.Cache Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
MemoryCacheManager::InsertSingleHash<T>() Method (T, CacheSettings, String)

Namespace: SimpleEssentials.Cache
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#

```csharp
public void InsertSingleHash<T>(
    T data,
    CacheSettings cacheSettings,
    string fieldKey
)
```

VB

```vbnet
Public Sub InsertSingleHash(Of T) ( _
    data As T, _
    cacheSettings As CacheSettings, _
    fieldKey As String _
)
```

C++

```cpp
public:

    template<typename T>
    void InsertSingleHash(
        T data,
        CacheSettings^ cacheSettings,
        String^ fieldKey
    )
```

Type Parameters

T

Parameters

data

Type: T

cacheSettings

Type: SimpleEssentials.Cache::CacheSettings

fieldKey

Type: String
See Also

MemoryCacheManager Class
InsertSingleHash Overload
SimpleEssentials.Cache Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
MemoryCacheManager::Update<T> Method

Namespace: SimpleEssentials.Cache
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#

VB

C++

public void Update<T>(
    T data,
    CacheSettings cacheSettings
)

Public Sub Update(Of T) ( _
    data As T, _
    cacheSettings As CacheSettings _
)

public:
    generic<typename T>
    void Update(
        T data,
        CacheSettings^ cacheSettings
    )

**Type Parameters**

T

**Parameters**

data
    Type: T

cacheSettings
    Type: SimpleEssentials.Cache..::.CacheSettings
See Also

MemoryCacheManager Class
SimpleEssentials.Cache Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
MemoryCacheManager..:..UpdateHa
<('T'>)> Method

Namespace:  SimpleEssentials.Cache
Assembly:  SimpleEssentials (in SimpleEssentials.dll)
## Syntax

### C#

```csharp
public void UpdateHash<T>(
    IEnumerable<T> data,
    CacheSettings cacheSettings
)
```

### VB

```vb
Public Sub UpdateHash(Of T) ( _
    data As IEnumerable(Of T), _
    cacheSettings As CacheSettings _
)
```

### C++

```cpp
public:
    template<typename T>
    void UpdateHash(
        IEnumerable<T> data,
        CacheSettings cacheSettings
    )
```

## Type Parameters

**T**

## Parameters

**data**

Type: `IEnumerable<T>`

**cacheSettings**

Type: `SimpleEssentials.Cache.CacheSettings`
See Also

MemoryCacheManager Class
SimpleEssentials.Cache Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
MemoryCacheManager..::.UpdateSingleHash<('T')> Method

Namespace: SimpleEssentials.Cache
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#

```csharp
public void UpdateSingleHash<T>(
    T data,
    CacheSettings cacheSettings
)
```

VB

```vbnet
Public Sub UpdateSingleHash(Of T) ( _
    data As T, _
    cacheSettings As CacheSettings _
)
```

C++

```cpp
public:
    generic<typename T>
    void UpdateSingleHash( 
        T data,
        CacheSettings^ cacheSettings
    )
```

Type Parameters

T

Parameters

data

Type: T

cacheSettings

Type: SimpleEssentials.Cache::<::CacheSettings
See Also

MemoryCacheManager Class
SimpleEssentials.Cache Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
RedisCacheManager Class

Namespace: SimpleEssentials.Cache
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#

Public Class RedisCacheManager _
    Implements ICacheManager

VB

Public Class RedisCacheManager _
    Implements ICacheManager

C++

public class RedisCacheManager : ICacheManager

public ref class RedisCacheManager : ICacheManager
Inheritance Hierarchy

Object
   SimpleEssentials.Cache....RedisCacheManager
See Also

RedisCacheManager Members
SimpleEssentials.Cache Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
RedisCacheManager Members

The RedisCacheManager type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Delete</strong></td>
<td>Determines whether the specified object is equal to the current object.</td>
</tr>
<tr>
<td><strong>DeleteHash</strong></td>
<td>(Inherited from Object.)</td>
</tr>
<tr>
<td><strong>DeleteSingleHash</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Equals(System.Object)</strong></td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)</td>
</tr>
<tr>
<td><strong>Finalize</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Get&lt;(Of &lt;&lt;'(T)&gt;)&gt;&gt;</strong></td>
<td></td>
</tr>
<tr>
<td><strong>GetData&lt;(Of &lt;&lt;'(T)&gt;)&gt;&gt;</strong></td>
<td></td>
</tr>
<tr>
<td><strong>GetHash&lt;(Of &lt;&lt;'(T)&gt;)&gt;&gt;</strong></td>
<td></td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Serves as the default hash function.</td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td>(Inherited from Object.)</td>
</tr>
<tr>
<td><strong>Insert&lt;(Of &lt;&lt;'(T)&gt;)&gt;&gt;</strong></td>
<td></td>
</tr>
<tr>
<td><strong>InsertHash&lt;(Of &lt;&lt;'(T)&gt;)&gt;&gt;</strong></td>
<td></td>
</tr>
<tr>
<td><strong>(IEnumerable&lt;(Of &lt;&lt;'(T)&gt;&gt;&gt;, CacheSettings)&gt;</strong></td>
<td></td>
</tr>
<tr>
<td><strong>InsertHash&lt;(Of &lt;&lt;'(T)&gt;)&gt;&gt;</strong></td>
<td></td>
</tr>
<tr>
<td><strong>(T, CacheSettings)&gt;</strong></td>
<td></td>
</tr>
<tr>
<td><strong>InsertSingleHash&lt;(Of &lt;&lt;'(T)&gt;)&gt;&gt;</strong></td>
<td></td>
</tr>
<tr>
<td><strong>(T, CacheSettings)&gt;</strong></td>
<td></td>
</tr>
<tr>
<td><strong>InsertSingleHash&lt;(Of &lt;&lt;'(T)&gt;)&gt;&gt;</strong></td>
<td></td>
</tr>
<tr>
<td><strong>(T, CacheSettings)&gt;</strong></td>
<td></td>
</tr>
</tbody>
</table>
**MemberwiseClone**

Creates a shallow copy of the current **Object**.
(Inherited from **Object**.)

**ToString**

Returns a string that represents the current object.
(Inherited from **Object**.)

**Update(Of <<'T>'>>)**

**UpdateHash(Of <<'T>'>>)**

**UpdateSingleHash(Of <<'T>'>>)**
See Also

RedisCacheManager Class
SimpleEssentials.Cache Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
RedisCacheManager Methods

The RedisCacheManager type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Delete</strong></td>
<td></td>
</tr>
<tr>
<td><strong>DeleteHash</strong></td>
<td></td>
</tr>
<tr>
<td><strong>DeleteSingleHash</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Equals(System.Object)</strong></td>
<td>Determines whether the specified object is equal to the current object. (Inherited from <strong>Object</strong>.)</td>
</tr>
<tr>
<td><strong>Finalize</strong></td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from <strong>Object</strong>.)</td>
</tr>
<tr>
<td><strong>Get&lt;(Of &lt;&lt;'(T)&gt;&gt;)&gt;&gt;</strong></td>
<td></td>
</tr>
<tr>
<td><strong>GetData&lt;(Of &lt;&lt;'(T)&gt;&gt;)&gt;&gt;</strong></td>
<td></td>
</tr>
<tr>
<td><strong>GetHash&lt;(Of &lt;&lt;'(T)&gt;&gt;)&gt;&gt;</strong></td>
<td></td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Serves as the default hash function. (Inherited from <strong>Object</strong>.)</td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Insert&lt;(Of &lt;&lt;'(T)&gt;&gt;)&gt;&gt;</strong></td>
<td></td>
</tr>
<tr>
<td><strong>InsertHash&lt;(Of &lt;&lt;'(T)&gt;&gt;)&gt;&gt;</strong></td>
<td></td>
</tr>
<tr>
<td><strong>(IEnumerable&lt;(Of &lt;&lt;'(T)&gt;&gt;), CacheSettings)</strong></td>
<td></td>
</tr>
<tr>
<td><strong>InsertHash&lt;(Of &lt;&lt;'(T)&gt;&gt;)&gt;&gt;</strong></td>
<td></td>
</tr>
<tr>
<td><strong>(T, CacheSettings)</strong></td>
<td></td>
</tr>
<tr>
<td><strong>InsertSingleHash&lt;(Of &lt;&lt;'(T)&gt;&gt;)(T, CacheSettings)</strong></td>
<td></td>
</tr>
<tr>
<td><strong>InsertSingleHash&lt;(Of &lt;&lt;'(T)&gt;&gt;)&gt;&gt;</strong></td>
<td></td>
</tr>
</tbody>
</table>
String)

MemberwiseClone

ToString

Create a shallow copy of the current **Object**.
(Inherited from **Object**.)

Returns a string that represents the current object.
(Inherited from **Object**.)

ToString

Update<(Of <<'(T)>>)

UpdateHash<(Of <<'(T)>>)

UpdateSingleHash<(Of <<'(T)>>)
See Also

RedisCacheManager Class
SimpleEssentials.Cache Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
RedisCacheManager:::Delete Method

Namespace: SimpleEssentials.Cache
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#

```csharp
public void Delete(
    CacheSettings cacheSettings
)
```

VB

```vbnet
Public Sub Delete ( _
    cacheSettings As CacheSettings _
)
```

C++

```cpp
public:
void Delete(
    CacheSettings^ cacheSettings
)
```

Parameters

`cacheSettings`

Type: `SimpleEssentials.Cache::CacheSettings`
See Also

RedisCacheManager Class
SimpleEssentials.Cache Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
RedisCacheManager..::..DeleteHash Method

Namespace: SimpleEssentials.Cache
Assembly: SimpleEssentials (in SimpleEssentials.dll)
**Syntax**

**C#**
```
public void DeleteHash(
    CacheSettings cacheSettings
)
```

**VB**
```
Public Sub DeleteHash (_
    cacheSettings As CacheSettings _
)
```

**C++**
```
public:
void DeleteHash(
    CacheSettings^ cacheSettings
)
```

**Parameters**

- **cacheSettings**
  Type: SimpleEssentials.Cache::CacheSettings
See Also

RedisCacheManager Class
SimpleEssentials.Cache Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
RedisCacheManager:::..DeleteSingleF
Method

Namespace:  SimpleEssentials.Cache
Assembly:  SimpleEssentials (in SimpleEssentials.dll)
Syntax

**C#**

```csharp
public void DeleteSingleHash(
    CacheSettings cacheSettings,
    string fieldKey
)
```

**VB**

```vb
Public Sub DeleteSingleHash (_
    cacheSettings As CacheSettings, _
    fieldKey As String _
)
```

**C++**

```cpp
public:
void DeleteSingleHash(
    CacheSettings^ cacheSettings,
    String^ fieldKey
)
```

**Parameters**

- `cacheSettings`
  - Type: `SimpleEssentials.Cache::CacheSettings`

- `fieldKey`
  - Type: `String`
See Also

RedisCacheManager Class
SimpleEssentials.Cache Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
RedisCacheManager..::.Get<(Of <('T)>)> Method

Namespace: SimpleEssentials.Cache
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#

```csharp
public T Get<T>(
    CacheSettings cacheSettings
)
```

VB

```vbnet
Public Function Get(Of T) ( _
    cacheSettings As CacheSettings _
) As T
```

C++

```cpp
public:
    generic<typename T>
    T Get(
        CacheSettings^ cacheSettings
    )
```

Type Parameters

T

Parameters

cacheSettings
    Type: SimpleEssentials.Cache::CacheSettings
See Also

RedisCacheManager Class
SimpleEssentials.Cache Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
RedisCacheManager...::..GetData<(Of <('T')>)> Method

Namespace: SimpleEssentials.Cache
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#

VB

C++

public T GetData<T>(
    CacheSettings cacheSettings,
    string fieldKey
)

Public Function GetData(Of T) ( _
    cacheSettings As CacheSettings, _
    fieldKey As String _
) As T

public:
    generic<typename T>
    T GetData(
        CacheSettings^ cacheSettings,
        String^ fieldKey
    )

Type Parameters

T

Parameters

cacheSettings
    Type: SimpleEssentials.Cache::..::CacheSettings

fieldKey
    Type: String
See Also

RedisCacheManager Class
SimpleEssentials.Cache Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
RedisCacheManager...::..GetHash<
<>("'T")>) Method

Namespace:  SimpleEssentials.Cache
Assembly:  SimpleEssentials (in SimpleEssentials.dll)
### Syntax

**C#**

```csharp
public IEnumerable<T> GetHash<T>(
    CacheSettings cacheSettings
)
```

**VB**

```vbnet
Public Function GetHash(Of T) ( _
    cacheSettings As CacheSettings _
) As IEnumerable(Of T)
```

**C++**

```cpp
public:
    generic<typename T>
    IEnumerable<T>& GetHash(
    CacheSettings^ cacheSettings
)
```

#### Type Parameters

- **T**

#### Parameters

- **cacheSettings**
  - Type: `SimpleEssentials.Cache...CacheSettings`
See Also

RedisCacheManager Class
SimpleEssentials.Cache Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
RedisCacheManager::<..GetList<Of <('T')>> Method

Namespace: SimpleEssentials.Cache
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#

```csharp
public IEnumerable<T> GetList<T>(
    CacheSettings cacheSettings
)
```

VB

```vbnet
Public Function GetList(Of T) ( _
    cacheSettings As CacheSettings _
) As IEnumerable(Of T)
```

C++

```cpp
public:
    generic<typename T>
    IEnumerable<T>^ GetList(
        CacheSettings^ cacheSettings
    )
```

Type Parameters

T

Parameters

cacheSettings
    Type: SimpleEssentials.Cache....CacheSettings
See Also

RedisCacheManager Class
SimpleEssentials.Cache Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
RedisCacheManager..::.GetSingleHas<T> Method

Namespace: SimpleEssentials.Cache
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#

```csharp
public T GetSingleHash<T>(
    CacheSettings cacheSettings,
    string fieldKey
)
```

VB

```vbnet
Public Function GetSingleHash(Of T) ( _
    cacheSettings As CacheSettings, _
    fieldKey As String _
) As T

public:
generic<typename T>
T GetSingleHash(  
    CacheSettings^ cacheSettings,  
    String^ fieldKey  
)
```

Type Parameters

T

Parameters

cacheSettings  
  Type: SimpleEssentials.Cache::::CacheSettings

fieldKey  
  Type: String
See Also

RedisCacheManager Class
SimpleEssentials.Cache Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
RedisCacheManager..::..Insert<(Of <('T')>)> Method

Namespace:  SimpleEssentials.Cache
Assembly:  SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#
public void Insert<T>(
    T data,
    CacheSettings cacheSettings
)

VB
Public Sub Insert(Of T) ( _
    data As T, _
    cacheSettings As CacheSettings _
)

C++
public:
    template<typename T>
    void Insert(
        T data,
        CacheSettings^ cacheSettings
    )

Type Parameters

T

Parameters

data
    Type: T

cacheSettings
    Type: SimpleEssentials.Cache..::..CacheSettings
See Also

RedisCacheManager Class
SimpleEssentials.Cache Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
RedisCacheManager...::..InsertHash
Method
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>InsertHash&lt;(Of &lt;&lt;'(T)&gt;&gt;)&gt;(IEnumerable&lt;(Of &lt;&lt;'(T)&gt;&gt;)&gt;&gt;, CacheSettings)</td>
<td></td>
</tr>
<tr>
<td>InsertHash&lt;(Of &lt;&lt;'(T)&gt;&gt;)&gt;(T, CacheSettings)</td>
<td></td>
</tr>
</tbody>
</table>
See Also

RedisCacheManager Class
RedisCacheManager Members
SimpleEssentials.Cache Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
RedisCacheManager..::..InsertHash<( Of <'T'>)>, CacheSettings)

Namespace: SimpleEssentials.Cache
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#  
void InsertHash<T>(
    IEnumerable<T> data,
    CacheSettings cacheSettings
)

VB  
Public Sub InsertHash(Of T)
    data As IEnumerable(Of T),
    cacheSettings As CacheSettings
End Sub

C++  
public:
    template<typename T>
    void InsertHash(
        IEnumerable<T> data,
        CacheSettings cacheSettings
    )

Type Parameters

T

Parameters

data
    Type: IEnumerable<(Of (Of<T>))>

cacheSettings
    Type: SimpleEssentials.Cache..:::CacheSettings
See Also

RedisCacheManager Class
InsertHash Overload
SimpleEssentials.Cache Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
RedisCacheManager...::..InsertHash<(<'T'>)> Method (T, CacheSettings)

Namespace: SimpleEssentials.Cache
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#

public void InsertHash<T>(
    T data,
    CacheSettings cacheSettings
)

VB

Public Sub InsertHash(Of T) ( _
    data As T, _
    cacheSettings As CacheSettings _
)

C++

public:
    template<typename T>
    void InsertHash(
        T data,
        CacheSettings^ cacheSettings
    )

Type Parameters

T

Parameters

data
    Type: T

cacheSettings
    Type: SimpleEssentials.Cache..::.CacheSettings
See Also

RedisCacheManager Class
InsertHash Overload
SimpleEssentials.Cache Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
RedisCacheManager:::..InsertSingleF
Method
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>InsertSingleHash&lt;(Of &lt;&lt;'(T)&gt;&gt;)(T, CacheSettings)</td>
<td></td>
</tr>
<tr>
<td>InsertSingleHash&lt;(Of &lt;&lt;'(T)&gt;&gt;)(T, CacheSettings, String)</td>
<td></td>
</tr>
</tbody>
</table>
See Also

RedisCacheManager Class
RedisCacheManager Members
SimpleEssentials.Cache Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
RedisCacheManager..::.InsertSingleF
<('T')> Method (T, CacheSettings)

Namespace: SimpleEssentials.Cache
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#

```csharp
public void InsertSingleHash<T>(
    T data,
    CacheSettings cacheSettings
)
```

VB

```vb
Public Sub InsertSingleHash(Of T) ( _
    data As T, _
    cacheSettings As CacheSettings _
)
```

C++

```cpp
public:
    template<typename T>
    void InsertSingleHash(
        T data,
        CacheSettings^ cacheSettings
    )
```

**Type Parameters**

T

**Parameters**

data
  Type: T

cacheSettings
  Type: SimpleEssentials.Cache::..::CacheSettings
See Also

RedisCacheManager Class
InsertSingleHash Overload
SimpleEssentials.Cache Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
RedisCacheManager..::..InsertSingleF<(<'T'>)> Method (T, CacheSettings, String)

Namespace:  SimpleEssentials.Cache
Assembly:  SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#  
public void InsertSingleHash<T>(
    T data,
    CacheSettings cacheSettings,
    string fieldKey
)

VB  
Public Sub InsertSingleHash(Of T) ( _
    data As T, _
    cacheSettings As CacheSettings, _
    fieldKey As String _
)

C++  
public:
    generic<
        typename T>
    void InsertSingleHash(
        T data,
        CacheSettings^ cacheSettings,
        String^ fieldKey
    )

Type Parameters

T

Parameters

data  
Type: T

cacheSettings  
Type: SimpleEssentials.Cache::Cache::CacheSettings

fieldKey  
Type: String
See Also

RedisCacheManager Class
InsertSingleHash Overload
SimpleEssentials.Cache Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
RedisCacheManager..::.Update<(Of <('T>')>) Method

Namespace: SimpleEssentials.Cache
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#

VB

C++

public void Update<T>(
    T data,
    CacheSettings cacheSettings
)

Public Sub Update(Of T) ( _
    data As T, _
    cacheSettings As CacheSettings _
)

public:
generic<typename T>
void Update(
    T data,
    CacheSettings^ cacheSettings
)

Type Parameters

T

Parameters

data
    Type: T

cacheSettings
    Type: SimpleEssentials.Cache...CacheSettings
See Also

RedisCacheManager Class
SimpleEssentials.Cache Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
RedisCacheManager..::.UpdateHash< T > Method

Namespace: SimpleEssentials.Cache
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#

```csharp
public void UpdateHash<T>(
    IEnumerable<T> data,
    CacheSettings cacheSettings
)
```

VB

```vbnet
Public Sub UpdateHash(Of T) (_
    data As IEnumerable(Of T), _
    cacheSettings As CacheSettings _
)
```

C++

```cpp
public:
    generic<typename T>
    void UpdateHash(
        IEnumerable<T> data,
        CacheSettings cacheSettings
    )
```

Type Parameters

T

Parameters

data
Type: `IEnumerable<Of <(T)>>`

cacheSettings
Type: `SimpleEssentials.Cache..::CacheSettings`
See Also

RedisCacheManager Class
SimpleEssentials.Cache Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
RedisCacheManager..::.UpdateSingle<('T')> Method

Namespace: SimpleEssentials.Cache
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#  
public void UpdateSingleHash<T>(
    T data,
    CacheSettings cacheSettings
)

VB  
Public Sub UpdateSingleHash(Of T) ( _
    data As T, _
    cacheSettings As CacheSettings _
)

C++

public:
    generic<typename T>
    void UpdateSingleHash(
        T data,
        CacheSettings^ cacheSettings
    )

Type Parameters

T

Parameters

data
    Type: T

cacheSettings
    Type: SimpleEssentials.Cache..:::CacheSettings
See Also

RedisCacheManager Class
SimpleEssentials.Cache Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
SessionCacheManager Class

Namespace:  SimpleEssentials.Cache
Assembly:  SimpleEssentials (in SimpleEssentials.dll)
Syntax

**C#**

```csharp
public class SessionCacheManager : ICacheManager
```

**VB**

```vbnet
Public Class SessionCacheManager
    Implements ICacheManager
```

**C++**

```cpp
public ref class SessionCacheManager : ICacheManager
```
Inheritance Hierarchy

Object
SimpleEssentials.Cache::SessionCacheManager
See Also

SessionCacheManager Members
SimpleEssentials.Cache Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
SessionCacheManager Members

The SessionCacheManager type exposes the following members.
Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Delete</td>
<td>Determines whether the specified object is equal to the current object.</td>
</tr>
<tr>
<td>DeleteHash</td>
<td></td>
</tr>
<tr>
<td>DeleteSingleHash</td>
<td></td>
</tr>
<tr>
<td>Equals(System.Object)</td>
<td>(Inherited from Object.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup</td>
</tr>
<tr>
<td>Get&lt;(Of &lt;&lt;'(T)&gt;&gt;)&gt;</td>
<td>operations before it is reclaimed by garbage collection.</td>
</tr>
<tr>
<td>GetData&lt;(Of &lt;&lt;'(T)&gt;&gt;)&gt;</td>
<td>(Inherited from Object.)</td>
</tr>
<tr>
<td>GetHash&lt;(Of &lt;&lt;'(T)&gt;&gt;)&gt;</td>
<td></td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function.</td>
</tr>
<tr>
<td>GetType</td>
<td>(Inherited from Object.)</td>
</tr>
<tr>
<td>Insert&lt;(Of &lt;&lt;'(T)&gt;&gt;)&gt;</td>
<td></td>
</tr>
<tr>
<td>InsertHash&lt;(Of &lt;&lt;'(T)&gt;&gt;)&gt;</td>
<td></td>
</tr>
<tr>
<td>(IEnumerable&lt;(Of &lt;&lt;'(T)&gt;&gt;&gt;, CacheSettings)</td>
<td></td>
</tr>
<tr>
<td>InsertHash&lt;(Of &lt;&lt;'(T)&gt;&gt;)&gt;</td>
<td></td>
</tr>
<tr>
<td>(T, CacheSettings)</td>
<td></td>
</tr>
<tr>
<td>GetList&lt;(Of &lt;&lt;'(T)&gt;&gt;)&gt;</td>
<td></td>
</tr>
<tr>
<td>GetSingleHash&lt;(Of &lt;&lt;'(T)&gt;&gt;)&gt;</td>
<td></td>
</tr>
<tr>
<td>Insert&lt;(Of &lt;&lt;'(T)&gt;&gt;)&gt;</td>
<td></td>
</tr>
<tr>
<td>InsertHash&lt;(Of &lt;&lt;'(T)&gt;&gt;)&gt;</td>
<td></td>
</tr>
<tr>
<td>(T, CacheSettings)</td>
<td></td>
</tr>
<tr>
<td>InsertSingleHash&lt;(Of &lt;&lt;'(T)&gt;&gt;) &gt; (T, CacheSettings)</td>
<td></td>
</tr>
<tr>
<td>InsertSingleHash&lt;(Of &lt;&lt;'(T)&gt;&gt;)&gt; (T, CacheSettings)</td>
<td></td>
</tr>
</tbody>
</table>
**MemberwiseClone**

Creates a shallow copy of the current **Object**.
(Inherited from **Object**.)

**ToString**

Returns a string that represents the current object.
(Inherited from **Object**.)

**Update<(Of <<'(T)>)>>**

**UpdateHash<(Of <<'(T)>)>>**

**UpdateSingleHash<(Of <<'(T)>)>>**
See Also

SessionCacheManager Class
SimpleEssentials.Cache Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
SessionCacheManager Methods

The SessionCacheManager type exposes the following members.
# Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Delete</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>DeleteHash</td>
<td></td>
</tr>
<tr>
<td>DeleteSingleHash</td>
<td></td>
</tr>
<tr>
<td>Equals(System.Object)</td>
<td></td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)</td>
</tr>
<tr>
<td>Get&lt;(Of &lt;&lt;'(T&gt;')&gt;&gt;)</td>
<td></td>
</tr>
<tr>
<td>GetData&lt;(Of &lt;&lt;'(T&gt;')&gt;&gt;)</td>
<td></td>
</tr>
<tr>
<td>GetHash&lt;(Of &lt;&lt;'(T&gt;')&gt;&gt;)</td>
<td></td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>Insert&lt;(Of &lt;&lt;'(T&gt;')&gt;&gt;)</td>
<td></td>
</tr>
<tr>
<td>InsertHash&lt;(Of &lt;&lt;'(T&gt;')&gt;&gt;)&gt;</td>
<td></td>
</tr>
<tr>
<td>(IEnumerable&lt;(Of &lt;&lt;'(T&gt;')&gt;&gt;), CacheSettings)</td>
<td></td>
</tr>
<tr>
<td>InsertHash&lt;(Of &lt;&lt;'(T&gt;')&gt;&gt;)&gt;</td>
<td></td>
</tr>
<tr>
<td>(T, CacheSettings)</td>
<td></td>
</tr>
<tr>
<td>InsertSingleHash&lt;(Of &lt;&lt;'(T&gt;')&gt;&gt;)(T, CacheSettings)</td>
<td></td>
</tr>
<tr>
<td>InsertSingleHash&lt;(Of &lt;&lt;'(T&gt;')&gt;&gt;)(T, CacheSettings,</td>
<td></td>
</tr>
</tbody>
</table>
**MemberwiseClone**

Creates a shallow copy of the current **Object**.
(Inherited from **Object**.)

**ToString**

Returns a string that represents the current object.
(Inherited from **Object**.)

**Update<T>>**

**UpdateHash<T>>**

**UpdateSingleHash<T>>**
See Also

SessionCacheManager Class
SimpleEssentials.Cache Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
SessionCacheManager..::.Delete Method

Namespace: SimpleEssentials.Cache
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#

```csharp
public void Delete(
    CacheSettings cacheSettings
)
```

VB

```vbnet
Public Sub Delete ( _
    cacheSettings As CacheSettings _
)
```

C++

```cpp
public:
void Delete(
    CacheSettings^ cacheSettings
)
```

Parameters

`cacheSettings`
Type: `SimpleEssentials.Cache:::CacheSettings`
See Also

SessionCacheManager Class
SimpleEssentials.Cache Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
SessionCacheManager..::..DeleteHash Method

Namespace: SimpleEssentials.Cache
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#

```csharp
public void DeleteHash(
    CacheSettings cacheSettings
)
```

VB

```vbnet
Public Sub DeleteHash ( _
    cacheSettings As CacheSettings _
)
```

C++

```cpp
public:
void DeleteHash(
    CacheSettings^ cacheSettings
)
```

Parameters

cacheSettings
  Type: SimpleEssentials.Cache::<>::CacheSettings
See Also

SessionCacheManager Class
SimpleEssentials.Cache Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
SessionCacheManager..::..DeleteSingleMethod

Namespace: SimpleEssentials.Cache
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#

```csharp
public void DeleteSingleHash(
    CacheSettings cacheSettings,
    string fieldKey
)
```

VB

```vbnet
Public Sub DeleteSingleHash (_
    cacheSettings As CacheSettings, _
    fieldKey As String _
)
```

C++

```cpp
public:
void DeleteSingleHash(
    CacheSettings^ cacheSettings,
    String^ fieldKey
)
```

Parameters

cacheSettings
Type: SimpleEssentials.Cache::CacheSettings

fieldKey
Type: String
See Also

- SessionCacheManager Class
- SimpleEssentials.Cache Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
SessionCacheManager..::.Get<('T')> Method

Namespace: SimpleEssentials.Cache
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#

public T Get<T>(
    CacheSettings cacheSettings
)

VB

Public Function Get(Of T) ( _
    cacheSettings As CacheSettings _
) As T

C++

public:

generic<typename T>
T Get(
    CacheSettings^ cacheSettings
)

Type Parameters

T

Parameters

cacheSettings
    Type: SimpleEssentials.Cache::CacheSettings
See Also

SessionCacheManager Class
SimpleEssentials.Cache Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
SessionCacheManager..::..GetData<T>() Method

Namespace: SimpleEssentials.Cache
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#
VB
C++

public T GetData<T>(
    CacheSettings cacheSettings,
    string fieldKey
)

Public Function GetData(Of T) ( _
    cacheSettings As CacheSettings, _
    fieldKey As String _
) As T

public:
    generic<typename T> T GetData(
        CacheSettings^ cacheSettings,
        String^ fieldKey
    )

Type Parameters

T

Parameters

cacheSettings
    Type: SimpleEssentials.Cache..::.CacheSettings

fieldKey
    Type: String
See Also

SessionCacheManager Class
SimpleEssentials.Cache Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
SessionCacheManager..::.GetHash<
<('T')>> Method

Namespace: SimpleEssentials.Cache
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#

```
public IEnumerable<T> GetHash<T>(
    CacheSettings cacheSettings)
```

VB

```
Public Function GetHash(Of T) ( _
    cacheSettings As CacheSettings _
) As IEnumerable(Of T)
```

C++

```
public:
    template<typename T>
    IEnumerable<T>^ GetHash(
        CacheSettings^ cacheSettings
    )
```

Type Parameters

T

Parameters

cacheSettings
    Type: SimpleEssentials.Cache::CacheSettings
See Also

SessionCacheManager Class
SimpleEssentials.Cache Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
SessionCacheManager..::.GetList(Of T) Method

Namespace: SimpleEssentials.Cache
Assembly: SimpleEssentials (in SimpleEssentials.dll)
## Syntax

### C#
```
public IEnumerable<T> GetList<T>(
    CacheSettings cacheSettings
)
```

### VB
```
Public Function GetList(Of T) ( _
    cacheSettings As CacheSettings _
) As IEnumerable(Of T)
```

### C++
```
public:
generic<
type T>
IEnumerable<T>^ GetList(
    CacheSettings^ cacheSettings
)
```

### Type Parameters

- **T**

### Parameters

- **cacheSettings**
  - Type: `SimpleEssentials.Cache..::..CacheSettings`
See Also

SessionCacheManager Class
SimpleEssentials.Cache Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
SessionCacheManager...::..GetSingleH<T> Method

Namespace: SimpleEssentials.Cache
Assembly: SimpleEssentials (in SimpleEssentials.dll)
**Syntax**

**C#**
```
public T GetSingleHash<T>(
    CacheSettings cacheSettings,
    string fieldKey
)
```

**VB**
```
Public Function GetSingleHash(Of T) ( _
    cacheSettings As CacheSettings, _
    fieldKey As String _
) As T
```

**C++**
```
public:
    generic<typename T>
T GetSingleHash(
    CacheSettings^ cacheSettings,
    String^ fieldKey
)
```

**Type Parameters**

**T**

**Parameters**

`cacheSettings`
Type: `SimpleEssentials.Cache:::CacheSettings`

`fieldKey`
Type: `String`
See Also

SessionCacheManager Class
SimpleEssentials.Cache Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
SessionCacheManager..::.Insert(Of <('T')> )> Method

Namespace: SimpleEssentials.Cache
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#
Public Sub Insert(Of T) ( _
    data As T, _
    cacheSettings As SimpleEssentials.Cache..::..CacheSettings _
)

VB
Public Sub Insert(Of T) ( _
    data As T, _
    cacheSettings As SimpleEssentials.Cache..::..CacheSettings _
)

C++
public: 
generic< typename T > 
void Insert(
    T data,
    SimpleEssentials.Cache..::..CacheSettings cacheSettings
)

Type Parameters

T

Parameters

data
    Type: T

cacheSettings
    Type: SimpleEssentials.Cache..::..CacheSettings
See Also

SessionCacheManager Class
SimpleEssentials.Cache Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
SessionCacheManager::InsertHash Method
### Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>InsertHash&lt;(Of &lt;&lt;'T'&gt;)&gt;(IEnumerable&lt;(Of &lt;&lt;'T'&gt;)&gt;, CacheSettings)&gt;</td>
<td></td>
</tr>
<tr>
<td>InsertHash&lt;(Of &lt;&lt;'T'&gt;)&gt;(T, CacheSettings)&gt;</td>
<td></td>
</tr>
</tbody>
</table>
See Also

SessionCacheManager Class
SessionCacheManager Members
SimpleEssentials.Cache Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
SessionCacheManager..::..InsertHash\<\(<'T'>\)\> Method
(IEnumerable\<Of\<\(<'T'>\)\>>>,,
CacheSettings)

Namespace: SimpleEssentials.Cache
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#

```csharp
public void InsertHash<T>(
    IEnumerable<T> data,
    CacheSettings cacheSettings
)
```

VB

```vbnet
Public Sub InsertHash(Of T) (_
    data As IEnumerable(Of T), _
    cacheSettings As CacheSettings _
)
```

C++

```cpp
public:
    template<typename T>
    void InsertHash(
        IEnumerable<T>^ data,
        CacheSettings^ cacheSettings
    )
```

Type Parameters

T

Parameters

data
    Type: IEnumerable<Of (Of<T>)>)

cacheSettings
    Type: SimpleEssentials.Cache..::..CacheSettings
See Also

SessionCacheManager Class
InsertHash Overload
SimpleEssentials.Cache Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
SessionCacheManager::InsertHash<T>(T, CacheSettings)

Namespace: SimpleEssentials.Cache
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#  
public void InsertHash<T>(
    T data,
    CacheSettings cacheSettings
)

VB  
Public Sub InsertHash(Of T)( _
    data As T, _
    cacheSettings As CacheSettings _
)

C++  
public:
    template<typename T>
    void InsertHash(
        T data,
        CacheSettings^ cacheSettings
    )

Type Parameters

T

Parameters

data
    Type: T

cacheSettings
    Type: SimpleEssentials.Cache..::.CacheSettings
See Also

SessionCacheManager Class
InsertHash Overload
SimpleEssentials.Cache Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
SessionCacheManager::InsertSingle Method
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>InsertSingleHash&lt;(Of &lt;&lt;'(T)&gt;&gt;)&gt;(T, CacheSettings)</code></td>
<td></td>
</tr>
<tr>
<td><code>InsertSingleHash&lt;(Of &lt;&lt;'(T)&gt;&gt;)&gt;(T, CacheSettings, String)</code></td>
<td></td>
</tr>
</tbody>
</table>
See Also

SessionCacheManager Class
SessionCacheManager Members
SimpleEssentials.Cache Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
SessionCacheManager..::.InsertSingle<T>()> Method (T, CacheSettings)

Namespace: SimpleEssentials.Cache
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#

public void InsertSingleHash<T>(
    T data,
    CacheSettings cacheSettings
)

VB

Public Sub InsertSingleHash(Of T) ( _
    data As T, _
    cacheSettings As CacheSettings _
)

C++

public:
    template<typename T>
    void InsertSingleHash( 
    T data, 
    CacheSettings^ cacheSettings 
)

Type Parameters

T

Parameters

data
    Type: T

cacheSettings
    Type: SimpleEssentials.Cache..::..CacheSettings
See Also

SessionCacheManager Class
InsertSingleHash Overload
SimpleEssentials.Cache Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
SessionCacheManager::InsertSingle<T>(T, CacheSettings, String)

Namespace: SimpleEssentials.Cache
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#

public void InsertSingleHash<T>(
    T data,
    CacheSettings cacheSettings,
    string fieldKey
)

VB

Public Sub InsertSingleHash(Of T) ( _
    data As T, _
    cacheSettings As CacheSettings, _
    fieldKey As String _
)

C++

public:

generic<
typename T>
void InsertSingleHash( _
    T data,
    CacheSettings^ cacheSettings,
    String^ fieldKey
)

Type Parameters

T

Parameters

data
    Type: T

cacheSettings
    Type: SimpleEssentials.Cache..::..CacheSettings

fieldKey
    Type: String
See Also

SessionCacheManager Class
InsertSingleHash Overload
SimpleEssentials.Cache Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
SessionCacheManager..::.Update(Of <('T')>)> Method

Namespace: SimpleEssentials.Cache
Assembly: SimpleEssentials (in SimpleEssentials.dll)
**Syntax**

C#  
Public void Update<T>(
    T data,
    CacheSettings cacheSettings
)

VB  
Public Sub Update(Of T) ( _
    data As T, _
    cacheSettings As CacheSettings _
)

C++  
public:
    template<typename T>
    void Update(
        T data,
        CacheSettings^ cacheSettings
    )

**Type Parameters**

T

**Parameters**

data
    Type: T

cacheSettings
    Type: SimpleEssentials.Cache::..::CacheSettings
See Also

SessionCacheManager Class
SimpleEssentials.Cache Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
SessionCacheManager..::.UpdateHasl
<('T')> Method

Namespace: SimpleEssentials.Cache
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#

```csharp
public void UpdateHash<T>(
    IEnumerable<T> data,
    CacheSettings cacheSettings
)
```

VB

```vbnet
Public Sub UpdateHash(Of T) (_
    data As IEnumerable(Of T), _
    cacheSettings As CacheSettings _
)
```

C++

```cpp
public:
    generic<typename T>
    void UpdateHash(
        IEnumerable<T>^ data,
        CacheSettings^ cacheSettings
    )
```

Type Parameters

T

Parameters

data
 Type: `IEnumerable`<Of `<(T)>`)

cacheSettings
 Type: `SimpleEssentials.Cache..::.CacheSettings`
See Also

SessionCacheManager Class
SimpleEssentials.Cache Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
SessionCacheManager..::.UpdateSing<br/>
<br/>
Namespace: SimpleEssentials.Cache<br/>
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#

```csharp
public void UpdateSingleHash<T>(
    T data,
    CacheSettings cacheSettings
)
```

VB

```vbnet
Public Sub UpdateSingleHash(Of T) ( _
    data As T, _
    cacheSettings As CacheSettings _
)
```

C++

```cpp
public:
    template<typename T>
    void UpdateSingleHash(
        T data,
        CacheSettings^ cacheSettings
    )
```

Type Parameters

T

Parameters

data
    Type: T

cacheSettings
    Type: `SimpleEssentials.Cache:::CacheSettings`
See Also

SessionCacheManager Class
SimpleEssentials.Cache Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
SessionObject<((Of <!T!>)>) Class

Namespace: SimpleEssentials.Cache
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#
VB
C++

public class SessionObject<T> : ISessionObject<T>

Public Class SessionObject(Of T) Implements ISessionObject(Of T)

generic<typename T>
public ref class SessionObject : ISessionObject<T>

Type Parameters

T
Inheritance Hierarchy

Object
SimpleEssentials.Cache:::SessionObject<Of (<'T'>)>
See Also

SessionObject<T> Members
SimpleEssentials.Cache Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
SessionObject<(Of '<T>')?>

Members

The SessionObject<(Of '<T>')?> type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>EQUALS(System.Object)</td>
<td>Determines whether the specified object is equal to the current object.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from Object.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from Object.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from Object.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from Object.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from Object.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from Object.)</td>
</tr>
</tbody>
</table>
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data</td>
<td></td>
</tr>
<tr>
<td>Expiration</td>
<td></td>
</tr>
</tbody>
</table>
See Also

SessionObject(Of(Of'T')>) Class
SimpleEssentials.Cache Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
SessionObject<(<'T'>)> Methods

The SessionObject<(<'T'>)> type exposes the following members.
# Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Equals(System.Object)</code></td>
<td>Determines whether the specified object is equal to the current object.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><code>Finalize</code></td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><code>GetHashCode</code></td>
<td>Serves as the default hash function.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><code>GetType</code></td>
<td>Gets the <code>Type</code> of the current instance.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><code>MemberwiseClone</code></td>
<td>Creates a shallow copy of the current <code>Object</code>.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><code>ToString</code></td>
<td>Returns a string that represents the current object.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <code>Object</code>.)</td>
</tr>
</tbody>
</table>
See Also

SessionObject<(Of (Of'))> Class
SimpleEssentials.Cache Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
SessionObject(Of (Of 'T)>)> Properties

The SessionObject(Of (Of 'T)>)> type exposes the following members.
Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data</td>
<td></td>
</tr>
<tr>
<td>Expiration</td>
<td></td>
</tr>
</tbody>
</table>
See Also

SessionObject<Of <(<'T'>)>> Class
SimpleEssentials.Cache Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
SessionObject<T>...Data Property

Namespace: SimpleEssentials.Cache
Assembly: SimpleEssentials (in SimpleEssentials.dll)
**Syntax**

**C#**

```csharp
public T Data { get; set; }
```

**VB**

```vbnet
Public Property Data As T
    Get
    Set
```

**C++**

```cpp
public:
    property T Data {
        T get ();
        void set (T value);
    }
```
See Also

SessionObject<Of '<T'>>() Class
SimpleEssentials.Cache Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
SessionObject<(Of (<'T'>))>...Expiration Property

Namespace: SimpleEssentials.Cache
Assembly: SimpleEssentials (in SimpleEssentials.dll)
**Syntax**

C#  
public Nullable<DateTime> Expiration { get; set; }

VB  
Public Property Expiration As Nullable(Of DateTime)
    Get
    Set

C++
public:
    property Nullable<DateTime^> Expiration {
        Nullable<DateTime^> get ();
        void set (Nullable<DateTime^> value);
    }
See Also

SessionObject<Of <(OfType)> > Class
SimpleEssentials.Cache Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
SimpleEssentials.Console Namespace
# Classes

<table>
<thead>
<tr>
<th>Class</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>😴Constants</td>
<td></td>
</tr>
</tbody>
</table>

*Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.*
Constants Class

Namespace:  SimpleEssentials.Console
Syntax

**C#**

```csharp
public static class Constants
```

**VB**

```vbnet
Public NotInheritable Class Constants
```

**C++**

```cpp
public ref class Constants abstract sealed
```
Inheritance Hierarchy

Object
  SimpleEssentials.Console:::Constants
See Also

Constants Members
SimpleEssentials.Console Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
Constants Members

The Constants type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>DbConnectionString</strong></td>
<td></td>
</tr>
</tbody>
</table>
See Also

Constants Class
SimpleEssentials.Console Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
Constants Methods

The Constants type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DbConnectionString</td>
<td></td>
</tr>
</tbody>
</table>
See Also

Constants Class
SimpleEssentials.Console Namespace

 Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
Constants..::..DbConnectionString

Method

Namespace:  SimpleEssentials.Console
Syntax

C#
VB
C++

public static string DbConnectionString()

Public Shared Function DbConnectionString As String

public:
static String^ DbConnectionString()
See Also

Constants Class
SimpleEssentials.Console Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
SimpleEssentials.Console.Models
Namespace
## Classes

<table>
<thead>
<tr>
<th>Class</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>🌟 CustomCampaign</td>
<td>Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.</td>
</tr>
</tbody>
</table>
CustomCampaign Class

Syntax

C#
VB
C++

[TableAttribute]
public class CustomCampaign

<typename>
Public Class CustomCampaign

[TableAttribute]
public ref class CustomCampaign
Inheritance Hierarchy

Object
SimpleEssentials.Console.Models..CustomCampaign
See Also

CustomCampaign Members
SimpleEssentials.Console.Models Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
CustomCampaign Members

The CustomCampaign type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Equals(System.Object)</code></td>
<td>Determines whether the specified object is equal to the current object. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><code>Finalize</code></td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><code>GetHashCode</code></td>
<td>Serves as the default hash function. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><code>GetType</code></td>
<td>Gets the <code>Type</code> of the current instance. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><code>MemberwiseClone</code></td>
<td>Creates a shallow copy of the current <code>Object</code>. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><code>ToString</code></td>
<td>Returns a string that represents the current object. (Inherited from <code>Object</code>.)</td>
</tr>
</tbody>
</table>
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CreateDate</td>
<td></td>
</tr>
<tr>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>Id</td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td></td>
</tr>
</tbody>
</table>
See Also

CustomCampaign Class
SimpleEssentials.Console.Models Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
CustomCampaign Methods

The CustomCampaign type exposes the following members.
# Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Equals(System.Object)</code></td>
<td>Determines whether the specified object is equal to the current object.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>.)</td>
</tr>
<tr>
<td><code>Finalize</code></td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>.)</td>
</tr>
<tr>
<td><code>GetHashCode</code></td>
<td>Serves as the default hash function.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>.)</td>
</tr>
<tr>
<td><code>GetType</code></td>
<td>Gets the <a href="https://docs.microsoft.com/en-us/dotnet/api/system.type">Type</a> of the current instance.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>.)</td>
</tr>
<tr>
<td><code>MemberwiseClone</code></td>
<td>Creates a shallow copy of the current <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>.)</td>
</tr>
<tr>
<td><code>ToString</code></td>
<td>Returns a string that represents the current object.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>.)</td>
</tr>
</tbody>
</table>
See Also

CustomCampaign Class
SimpleEssentials.Console.Models Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
CustomCampaign Properties

The CustomCampaign type exposes the following members.
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CreateDate</td>
<td></td>
</tr>
<tr>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>Id</td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td></td>
</tr>
</tbody>
</table>
See Also

CustomCampaign Class
SimpleEssentials.Console.Models Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
CustomCampaign..::..CreateDate Property

Syntax

C#

```csharp
public Nullable<DateTime> CreateDate { get; set; }
```

VB

```vbnet
Public Property CreateDate As Nullable(Of DateTime)
    Get
    Set
```

C++

```cpp
public:
property Nullable<DateTime^> CreateDate {
    Nullable<DateTime^> get ();
    void set (Nullable<DateTime^> value);
}
```
See Also

CustomCampaign Class
SimpleEssentials.Console.Models Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
CustomCampaign..::..Description Property

Syntax

**C#**

```csharp
public string Description { get; set; }
```

**VB**

```vbnet
Public Property Description As String
    Get
    Set
```

**C++**

```c++
public:
property String^ Description {
    String^ get ();
    void set (String^ value);
}
```
See Also

CustomCampaign Class
SimpleEssentials.Console.Models Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
CustomCampaign..::..Id Property

Syntax

C#
VB
C++

public int Id { get; set; }

Public Property Id As Integer
   Get
   Set

public:
property int^ Id {
    int^ get ();
    void set (int^ value);
}
See Also

CustomCampaign Class
SimpleEssentials.Console.Models Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
CustomCampaign...Name Property

Syntax

C#
VB
C++

public string Name { get; set; }

Public Property Name As String
    Get
    Set

public:
property String^ Name {
    String^ get ();
    void set (String^ value);
}
See Also

CustomCampaign Class
SimpleEssentials.Console.Models Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
SimpleEssentials.DataProvider
Namespace
## Classes

<table>
<thead>
<tr>
<th>Class</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DbDataProvider</td>
<td>SQL Database Data Provider.</td>
</tr>
</tbody>
</table>
## Interfaces

<table>
<thead>
<tr>
<th>Interface</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IDataProvider</td>
<td></td>
</tr>
<tr>
<td>IDbProvider</td>
<td>Database Provider Interface</td>
</tr>
<tr>
<td>IFileProvider</td>
<td></td>
</tr>
</tbody>
</table>

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
DbDataProvider Class

SQL Database Data Provider.

Namespace: SimpleEssentials.DataProvider
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#

public class DbDataProvider : IDbProvider

VB

Public Class DbDataProvider
    Implements IDbProvider

C++

public ref class DbDataProvider : IDbProvider
Inheritance Hierarchy

Object
SimpleEssentials.DataProvider......DbDataProvider
See Also

DbDataProvider Members
SimpleEssentials.DataProvider Namespace
SimpleEssentials.DataProvider..::..IDbProvider
SimpleEssentials.DataProvider..::..IDbProvider

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
DbDataProvider Members

The `DbDataProvider` type exposes the following members.
Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DbDataProvider</td>
<td>Initializes a new instance of the DbDataProvider class.</td>
</tr>
<tr>
<td>DbDataProvider(IDataStore, ICacheManager)</td>
<td>Initializes a new instance of the DbDataProvider class.</td>
</tr>
</tbody>
</table>
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>BulkInsert</strong>(Of &lt;&lt;'(T)&gt;&gt;)</td>
<td>TODO. Currently Not working</td>
</tr>
<tr>
<td><strong>CreateTable</strong>(Of &lt;&lt;'(T)&gt;&gt;)</td>
<td>Creates a table in the database.</td>
</tr>
<tr>
<td><strong>Delete</strong>(Of &lt;&lt;'(T)&gt;&gt;)</td>
<td>Deleted the row in the table.</td>
</tr>
<tr>
<td><strong>Equals(System.Object)</strong></td>
<td>Determines whether the specified object is equal to the current object. (Inherited from <strong>Object</strong>.)</td>
</tr>
<tr>
<td><strong>Execute</strong></td>
<td>Executes the specified SQL.</td>
</tr>
<tr>
<td><strong>ExecuteScalar</strong></td>
<td>Executes the scalar.</td>
</tr>
<tr>
<td><strong>Finalize</strong></td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from <strong>Object</strong>.)</td>
</tr>
<tr>
<td><strong>Get</strong>(Of &lt;&lt;'(T)&gt;&gt;)</td>
<td>Get a row from the database by ID.</td>
</tr>
<tr>
<td><strong>Get</strong>(Of &lt;&lt;'(T)&gt;&gt;)</td>
<td>Get all rows from a table.</td>
</tr>
<tr>
<td><strong>Get</strong>(Of &lt;&lt;'(T)&gt;&gt;)</td>
<td>Get all rows that match the expression given</td>
</tr>
<tr>
<td><strong>GetByParameters</strong>(Of &lt;&lt;'(T)&gt;&gt;)</td>
<td>Gets all rows by a sql statement and optional parameters.</td>
</tr>
<tr>
<td><strong>GetByType</strong>(Of &lt;&lt;'(T)&gt;&gt;)</td>
<td>Get all rows from a table.</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Serves as the default hash function. (Inherited from <strong>Object</strong>.)</td>
</tr>
<tr>
<td><strong>GetMultiMap</strong>(Of &lt;&lt;'(T, T2)&gt;&gt;)</td>
<td>Builds an object that references multiple tables/rows.</td>
</tr>
<tr>
<td><strong>GetMultiMap</strong>(Of &lt;&lt;'(T, T2, T3)&gt;&gt;)</td>
<td>Gets the multi map.</td>
</tr>
</tbody>
</table>
T4>)>>>(String, Func<(Of <<'(T, T2, T3, T4, T5, T6)>>) Object, String, CacheSettings)
GetMultiMap<(Of <<'(T, T2, T3, T4, T5, T6)>>)>(String, Func<(Of <<'(T, T2, T3, T4, T5, T6, T>)>>) Object, String, CacheSettings)
GetMultiMap<(Of <<'(T, T2, T3, T4, T5, T6)>>)>(String, Func<(Of <<'(T, T2, T3, T4, T5, T6, T>)>>) Object, String, CacheSettings)

GetMultiMap<(Of <<'(T, T2, T3, T4, T5)>>)>(String, Func<(Of <<'(T, T2, T3, T4, T5, T6)>>) Object, String, CacheSettings)
GetMultiMap<(Of <<'(T, T2, T3, T4, T5)>>)>(String, Func<(Of <<'(T, T2, T3, T4, T5, T6)>>) Object, String, CacheSettings)
GetMultiMap<(Of <<'(T, T2, T3, T4, T5)>>)>(String, Func<(Of <<'(T, T2, T3, T4, T5, T6)>>) Object, String, CacheSettings)

GetType

Insert<(Of <<'(T)>>)>
InsertAndReturnId<(Of <<'(T)>>)>(T, CacheSettings)
InsertAndReturnId<(Of <<'(T)>>)>(String, T, CacheSettings)
InsertList<(Of <<'(T)>>)>(IEnumerable<(Of <<'(T)>>)>, CacheSettings)
InsertList<(Of <<'(T)>>)>(IEnumerable<(Of <<'(T)>>)>, String, CacheSettings)

MemberwiseClone

ToString

Update<(Of <<'(T)>>)>

Gets the multi map.

Gets the multi map.

Gets the multi map.

Gets the Type of the current instance.
(Inherited from Object.)
Inserts a strongly typed object into a table.
Inserts an object into a table and then return its ID.
Inserts an object into a table and then return its ID.
Insert a list of objects into a table.
Insert a list of objects into a table.

Creates a shallow copy of the current Object.
(Inherited from Object.)
Returns a string that represents the current object.
(Inherited from Object.)
Updates te row in the table by the object passed.
See Also

DbDataProvider Class
SimpleEssentials.DataProvider Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
DbDataProvider Constructor
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DbDataProvider</td>
<td>Initializes a new instance of the DbDataProvider class.</td>
</tr>
<tr>
<td>DbDataProvider(IDDataStore, ICacheManager)</td>
<td>Initializes a new instance of the DbDataProvider class.</td>
</tr>
</tbody>
</table>
See Also

- DbDataProvider Class
- DbDataProvider Members
- SimpleEssentials.DataProvider Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
DbDataProvider Constructor

Initializes a new instance of the DbDataProvider class.

**Namespace:**  SimpleEssentials.DataProvider  
**Assembly:**  SimpleEssentials (in SimpleEssentials.dll)
## Syntax

### C#
```csharp
public DbDataProvider()
```

### VB
```vbnet
Public Sub New
```

### C++
```cpp
public:
DbDataProvider()
```
See Also

DbDataProvider Class
DbDataProvider Overload
SimpleEssentials.DataProvider Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
DbDataProvider Constructor (IDataStore, ICacheManager)

Initializes a new instance of the DbDataProvider class.

Namespace: SimpleEssentials.DataProvider
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#
VB
C++

public DbDataProvider(
    IDataStore dataStore,
    ICacheManager cacheManager
)

Public Sub New (_
    dataStore As IDataStore, _
    cacheManager As ICacheManager _
)

public:
DbDataProvider(
    IDataStore^ dataStore,
    ICacheManager^ cacheManager
)

Parameters

dataStore
    Type: SimpleEssentials.DataStore:::IDataStore
    The data store.

cacheManager
    Type: SimpleEssentials.Cache:::ICacheManager
    The cache manager.
See Also

DbDataProvider Class  
DbDataProvider Overload  
SimpleEssentials.DataProvider Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
DbDataProvider Methods

The DbDataProvider type exposes the following members.
# Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BulkInsert&lt;&lt;T&gt;&gt;</td>
<td>TODO. Currently Not working</td>
</tr>
<tr>
<td>CreateTable&lt;&lt;T&gt;&gt;</td>
<td>Creates a table in the database.</td>
</tr>
<tr>
<td>Delete&lt;&lt;T&gt;&gt;</td>
<td>Deleted the row in the table.</td>
</tr>
<tr>
<td>Equals(System.Object)</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>Execute</td>
<td>Executes the specified SQL.</td>
</tr>
<tr>
<td>ExecuteScalar</td>
<td>Executes the scalar.</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>Get&lt;&lt;T&gt;&gt;&gt;(Object, CacheSettings)</td>
<td>Get a row from the database by ID.</td>
</tr>
<tr>
<td>Get&lt;&lt;T&gt;&gt;&gt;(CacheSettings)</td>
<td>Get all rows from a table.</td>
</tr>
<tr>
<td>Get&lt;&lt;T&gt;&gt;&gt;(Expression&lt;&lt;Of &lt;&lt;(Func&lt;&lt;(T, Boolean)&gt;&gt;)&gt;&gt;), CacheSettings)</td>
<td>Get all rows that match the expression given</td>
</tr>
<tr>
<td>GetByParameters&lt;&lt;T&gt;&gt;</td>
<td>Gets all rows by a sql statement and optional parameters.</td>
</tr>
<tr>
<td>GetByType&lt;&lt;T&gt;&gt;</td>
<td>Get all rows from a table.</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>GetMultiMap&lt;&lt;T, T2&gt;&gt;(String, Func&lt;&lt;(Of &lt;&lt;(T, T2, T3)&gt;&gt;&gt;), Object, String, CacheSettings)</td>
<td>Builds an object that references multiple tables/rows.</td>
</tr>
<tr>
<td>GetMultiMap&lt;&lt;T, T2&gt;&gt;(String, Func&lt;&lt;(Of &lt;&lt;(T, T2, T3)&gt;&gt;&gt;), Object, String, CacheSettings)</td>
<td>Gets the multi map.</td>
</tr>
</tbody>
</table>
**GetMultiMap**〈(Of 〈(Of 〈(T, T2, T3, T4, T5)〉), Object, String, CacheSettings)〉,(String, Func〈(Of 〈(T, T2, T3, T4, T5, T6)〉), Object, String, CacheSettings)〉) Gets the multi map.

**GetMultiMap**〈(Of 〈(T, T2, T3, T4, T5, T6)〉)〉(String, Func〈(Of 〈(T, T2, T3, T4, T5, T6, T)〉), Object, String, CacheSettings)〉 Gets the multi map.

**GetMultiMap**〈(Of 〈(T, T2, T3, T4, T5, T6)〉)〉(String, Func〈(Of 〈(T, T2, T3, T4, T5, T6, T)〉), Object, String, CacheSettings)〉 Gets the multi map.

**GetType**

**Insert**〈(Of 〈(T)〉)〉

**InsertAndReturnId**〈(Of 〈(T)〉)〉(T, CacheSettings) Inserts an object into a table and then return its ID.

**InsertAndReturnId**〈(Of 〈(T)〉)〉(String, T, CacheSettings) Inserts an object into a table and then return its ID.

**InsertList**〈(Of 〈(T)〉)〉

**InsertList**〈(Of 〈(T)〉)〉(IEnumerable〈(Of 〈(T)〉)〉, CacheSettings) Insert a list of objects into a table.

**InsertList**〈(Of 〈(T)〉)〉(IEnumerable〈(Of 〈(T)〉)〉, String, CacheSettings) Insert a list of objects into a table.

**MemberwiseClone**

**ToString**

**Update**〈(Of 〈(T)〉)〉

Gets the **Type** of the current instance.
(Inherited from **Object**.)

Inserts a strongly typed object into a table.

Inserts an object into a table and then return its ID.

Inserts an object into a table and then return its ID.

Insert a list of objects into a table.

Insert a list of objects into a table.

Creates a shallow copy of the current **Object**.
(Inherited from **Object**.)

Returns a string that represents the current object.
(Inherited from **Object**.)

Updates te row in the table by the object passed.
See Also

DbDataProvider Class
SimpleEssentials.DataProvider Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
DbDataProvide...::..BulkInsert<(Of <('T>')>)> Method

TODO. Currently Not working

Namespace: SimpleEssentials.DataProvider
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#

public void BulkInsert<T>(
    IEnumerable<T> data,
    string tableName,
    CacheSettings cacheSettings
)

VB

Public Sub BulkInsert(Of T) (_
    data As IEnumerable(Of T), _
    tableName As String, _
    cacheSettings As CacheSettings _
)

C++

public:

    template<typename T>
    void BulkInsert(
        IEnumerable<T> data,
        String tableName,
        CacheSettings cacheSettings
    )

Type Parameters

T

Parameters

data

Type: IEnumerable<Of ((T))>
The data.

tableName

Type: String
Name of the table.
cacheSettings

Type: SimpleEssentials.Cache.CacheSettings

The cache settings.
See Also

DbDataProvider Class
SimpleEssentials.DataProvider Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
DbDataProvider... CreateTable(Of <('T')>) Method

Creates a table in the database.

Namespace: SimpleEssentials.DataProvider
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#  
public bool CreateTable<T>()

VB  
Public Function CreateTable(Of T) As Boolean

C++  
public:  
generic<typename T>  
bool^ CreateTable()  

Type Parameters

T

Return Value

True is successfull, false if table exists already.
See Also

DbDataProvider Class
SimpleEssentials.DataProvider Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
DbDataProvider:::Delete<(<'T'>)>) Method

Deleted the row in the table.

Namespace: SimpleEssentials.DataProvider
Assembly: SimpleEssentials (in SimpleEssentials.dll)
### Syntax

**C#**

```csharp
public bool Delete<T>(
    T data,
    CacheSettings cacheSettings,
    string fieldKey
)
```

**VB**

```vbnet
Public Function Delete(Of T) ( _
    data As T, _
    cacheSettings As CacheSettings, _
    fieldKey As String _
) As Boolean
```

**C++**

```cpp
public: bool^ Delete<T>(
    T data,
    CacheSettings^ cacheSettings,
    String^ fieldKey
)
```

### Type Parameters

**T**

### Parameters

**data**

Type: T
The data.

**cacheSettings**

Type: SimpleEssentials.Cache::CacheSettings
The cache settings.
fieldKey
  Type: String
  The field key.

**Return Value**

True if successful, false if not
See Also

DbDataProvider Class
SimpleEssentials.DataProvider Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
DbDataProvider::Execute Method

Executes the specified SQL.

Namespace: SimpleEssentials.DataProvider
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#

```csharp
public int Execute(
    string sql,
    Object param,
    CacheSettings cacheSettings,
    bool invalidateCache
)
```

VB

```vbnet
Public Function Execute (_
    sql As String, _
    param As Object, _
    cacheSettings As CacheSettings, _
    invalidateCache As Boolean _
) As Integer
```

C++

```cpp
public:
    int^ Execute(
        String^ sql,
        Object^ param,
        CacheSettings^ cacheSettings,
        bool^ invalidateCache
    )
```

Parameters

sql
Type: String
The SQL.

param
Type: Object
The parameter.

cacheSettings
Type: SimpleEssentials.Cache..CacheSettings
The cache settings.
invalidateCache
   Type: **Boolean**
   if set to true [invalidate cache].

**Return Value**

TODO
See Also

DbDataProvider Class
SimpleEssentials.DataProvider Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
DbDataProvider::..ExecuteScalar Method

Executes the scalar.

**Namespace:** SimpleEssentials.DataProvider  
**Assembly:** SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#
VB
C++

public int ExecuteScalar(
    string sql,
    Object param,
    CacheSettings cacheSettings,
    bool invalidateCache
)

Public Function ExecuteScalar ( _
    sql As String, _
    param As Object, _
    cacheSettings As CacheSettings, _
    invalidateCache As Boolean _
) As Integer

public:
int^ ExecuteScalar(
    String^ sql,
    Object^ param,
    CacheSettings^ cacheSettings,
    bool^ invalidateCache
)

Parameters

sql
    Type: String
    The SQL.

param
    Type: Object
    The parameter.

cacheSettings
    Type: SimpleEssentials.Cache..:::CacheSettings
    The cache settings.
invalidateCache
    Type: Boolean
    if set to true [invalidate cache].

Return Value

TODO
See Also

DbDataProvider Class
SimpleEssentials.DataProvider Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Get&lt;(Of &lt;&lt;'(T)&gt;&gt;)(Object, CacheSettings)</code></td>
<td>Get a row from the database by ID.</td>
</tr>
<tr>
<td><code>Get&lt;(Of &lt;&lt;'(T)&gt;&gt;)(CacheSettings)</code></td>
<td>Get all rows from a table.</td>
</tr>
<tr>
<td><code>Get&lt;(Of &lt;&lt;'(T)&gt;&gt;)(Expression&lt;(Of &lt;&lt;'(Func&lt;(Of &lt;&lt;'(T, Boolean)&gt;&gt;)&gt;&gt;)&gt;, CacheSettings)</code></td>
<td>Get all rows that match the expression given</td>
</tr>
</tbody>
</table>
See Also

DbDataProvider Class
DbDataProvider Members
SimpleEssentials.DataProvider Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
DbDataProvider...::..Get<(Of
<('T')>)> Method (Object,
CacheSettings)

Get a row from the database by ID.

Namespace: SimpleEssentials.DataProvider
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#  

```csharp
public T Get<T>(
    Object id,
    CacheSettings cacheSettings
)
```

VB  

```vbnet
Public Function Get(Of T) ( _
    id As Object, _
    cacheSettings As CacheSettings _
) As T
```

C++  

```cpp
public:
    template<typename T>
    T Get(
    Object^ id,
    CacheSettings^ cacheSettings
    )
```

**Type Parameters**

T

**Parameters**

id  
Type: Object  
The identifier.

cacheSettings  
Type: SimpleEssentials.Cache..::.CacheSettings  
The cache settings.

**Return Value**
Strongly typed object
See Also

DbDataProvider Class
Get Overload
SimpleEssentials.DataProvider Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
DbDataProvider..::.Get<(Of <('T')>)> Method (CacheSettings)

Get all rows from a table.

**Namespace:** SimpleEssentials.DataProvider  
**Assembly:** SimpleEssentials (in SimpleEssentials.dll)
### Syntax

#### C#

```csharp
public IEnumerable<T> Get<T>(
    CacheSettings cacheSettings
)
```

#### VB

```vbnet
Public Function Get(Of T) ( _
    cacheSettings As CacheSettings _
) As IEnumerable(Of T)
```

#### C++

```cpp
public:
    template<typename T>
    IEnumerable<T>^ Get(
        CacheSettings^ cacheSettings
    )
```

### Type Parameters

- **T**

### Parameters

- **cacheSettings**
  
  Type: [SimpleEssentials.Cache::CacheSettings](#)
  The cache settings.

### Return Value

- A list of strongly typed objects
See Also

DbDataProvider Class
Get Overload
SimpleEssentials.DataProvider Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
DbDataProvider...:::.Get<(Of <('T')>)> Method (Expression<(Of <('Func<(Of <('T, Boolean')>)>>)>>, CacheSettings)

Get all rows that match the expression given

**Namespace:**  SimpleEssentials.DataProvider
**Assembly:**  SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#
**IEnumerator**<T> Get<T>(
    **Expression**<Func<T, **bool**>> expression,
    **CacheSettings** cacheSettings
)

VB
Public Function Get(Of T) ( _
    expression As **Expression**(Of Func(Of T, **Boolean**)), _
    cacheSettings As **CacheSettings** _
) As **IEnumerator**(Of T)

C++
public:
  generic<typename T>
  **IEnumerator**<T>^ Get(
    **Expression**<Func<T, **bool**>>^ expression,
    **CacheSettings**^ cacheSettings
  )

Type Parameters

T

Parameters

expression
  Type: **Expression**<(Of <('Func<(Of <('T, **Boolean**))>>)>)>
  The expression.

cacheSettings
  Type: **SimpleEssentials.Cache..::.CacheSettings**
  The cache settings.

Return Value
Gets a list of strongly typed objects
See Also

DbDataProvider Class
Get Overload
SimpleEssentials.DataProvider Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
DbDataProvider...:::GetProperty Parameters <('T')> Method

Gets all rows by a sql statement and optional parameters.

Namespace: SimpleEssentials.DataProvider
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#

VB

C++

public IEnumerable<T> GetByParameters<T>(
    string sql,
    Object param,
    CacheSettings cacheSettings
)

Public Function GetByParameters(Of T) ( _
    sql As String, _
    param As Object, _
    cacheSettings As CacheSettings _
) As IEnumerable(Of T)

public:
    generic<typename T>
    IEnumerable<T>^ GetByParameters(
        String^ sql,
        Object^ param,
        CacheSettings^ cacheSettings
    )

Type Parameters

T

Parameters

sql
    Type: String
    The SQL.

param
    Type: Object
    The parameter.
cacheSettings
   Type: SimpleEssentials.Cache.CacheSettings
   The cache settings.

**Return Value**

IEnumerable<T>.
See Also

DbDataProvider Class
SimpleEssentials.DataProvider Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
DbDataProvider..::..GetByType<(Of <('T')>)) Method

Get all rows from a table.

Namespace: SimpleEssentials_DataProvider
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#

```csharp
public IEnumerable<T> GetByType<T>(
    CacheSettings cacheSettings
)
```

VB

```vbnet
Public Function GetByType(Of T) ( _
    cacheSettings As CacheSettings _
) As IEnumerable(Of T)
```

C++

```cpp
public:
    template<typename T>
    IEnumerable<T>^ GetByType(  
        CacheSettings^ cacheSettings  
    )
```

Type Parameters

T

Parameters

cacheSettings
Type: `SimpleEssentials.Cache..::..CacheSettings`
The cache settings.

Return Value

List of strongly typed objects
See Also

DbDataProvider Class
SimpleEssentials.DataProvider Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
DbDataProvider...:..GetMultiMap<(O<br><(<'T, T2>>)> Method (String,<br>Func<(Of <((<'T, T2, T>>))>, Object,<br>String, CacheSettings)<br>

Builds an object that references multiple tables/rows.

Syntax

C#

```csharp
public IEnumerable<T> GetMultiMap<T, T2>(
    string sql,
    Func<T, T2, T> func,
    Object param,
    string splitOn,
    CacheSettings cacheSettings
)
```

VB

```vbnet
Public Function GetMultiMap(Of T, T2)(
    _
    sql As String, _
    func As Func(Of T, T2, T), _
    param As Object, _
    splitOn As String, _
    cacheSettings As CacheSettings _
) As IEnumerable(Of T)
```

C++

```cpp
public:
    template<typename T, typename T2>
    IEnumerable<T>^ GetMultiMap(  
        String^ sql,
        Func<T, T2, T>^ func,
        Object^ param,
        String^ splitOn,
        CacheSettings^ cacheSettings
    )
```

Type Parameters

T
T2

The type of the 2.

Parameters

sql
The SQL.

func
  Type: Func<(Of <(<'T, T2, T>>)>
  The function.

param
  Type: Object
  The parameter.

splitOn
  Type: String
  The split on.

cacheSettings
  Type: SimpleEssentials.Cache..:::CacheSettings
  The cache settings.

**Return Value**

Strongly typed object whose data is mapped for you
See Also

DbDataProvider Class
GetMultiMap Overload
SimpleEssentials.DataProvider Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
DbDataProvider::GetMultiMap<(Of <'T, T2, T3>)> Method (String, Func<(Of <'T, T2, T3, T>), Object, String, CacheSettings>)

Gets the multi map.

Namespace: SimpleEssentials.DataProvider
Assembly: SimpleEssentials (in SimpleEssentials.dll)
**Syntax**

C#  
VB  
C++

```csharp
public IEnumerable<T> GetMultiMap<T, T2, T3>(
    string sql,
    Func<T, T2, T3, T> func,
    Object param,
    string splitOn,
    CacheSettings cacheSettings
)
```

```vbnet
Public Function GetMultiMap(Of T, T2, T3) ( _
    sql As String, _
    func As Func(Of T, T2, T3, T), _
    param As Object, _
    splitOn As String, _
    cacheSettings As CacheSettings _
) As IEnumerable(Of T)
```

```csharp
public:
generic<tyepname T, typename T2, typename T3>
IEnumerable<T>^ GetMultiMap(
    String^ sql,
    Func<T, T2, T3, T>^ func,
    Object^ param,
    String^ splitOn,
    CacheSettings^ cacheSettings
)
```

**Type Parameters**

- **T**
- **T2**
  The type of the 2.
- **T3**
  The type of the 3.

**Parameters**
sql
   Type: String
   The SQL.

func
   Type: Func<(Of <('T, T2, T3, T)>))
   The function.

param
   Type: Object
   The parameter.

splitOn
   Type: String
   The split on.

cacheSettings
   Type: SimpleEssentials.Cache..::..CacheSettings
   The cache settings.

**Return Value**

IEnumerable<T>.
See Also

DbDataProvider Class
GetMultiMap Overload
SimpleEssentials.DataProvider Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
DbDataProvider:::..GetMultiMap<\(\text{Of } \langle\langle 'T, T2, T3, T4\rangle\rangle\rangle\) Method
\(\text{String, Func\langle\langle \text{Of } \langle\langle 'T, T2, T3, T4, T\rangle\rangle\rangle\rangle, Object, String, CacheSettings}\)

Gets the multi map.

Namespace: SimpleEssentials.DataProvider
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#

VB

C++

public IEnumerable<T> GetMultiMap<T, T2, T3, T4>(
    string sql,
    Func<T, T2, T3, T4, T> func,
    Object param,
    string splitOn,
    CacheSettings cacheSettings
)

Public Function GetMultiMap(Of T, T2, T3, T4) ( _
    sql As String, _
    func As Func(Of T, T2, T3, T4, T), _
    param As Object, _
    splitOn As String, _
    cacheSettings As CacheSettings _
) As IEnumerable(Of T)

public:
    generic<typename T, typename T2, typename T3, typename T4>
    IEnumerable<T^> GetMultiMap(
        String^ sql,
        Func<T, T2, T3, T4, T^> func,
        Object^ param,
        String^ splitOn,
        CacheSettings^ cacheSettings
    )

Type Parameters

T
    The type of the 1.
T2
    The type of the 2.
T3
    The type of the 3.
T4
    The type of the 4.
**Parameters**

**sql**
Type: **String**
The SQL.

**func**
Type: Func<(Of <*> T, T2, T3, T4, T)>
The function.

**param**
Type: **Object**
The parameter.

**splitOn**
Type: **String**
The split on.

**cacheSettings**
Type: **SimpleEssentials.Cache..::.CacheSettings**
The cache settings.

**Return Value**

IEnumerable<T>.
See Also

DbDataProvider Class
GetMultiMap Overload
SimpleEssentials.DataProvider Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
DbDataProvider...:::GetMultiMap<(Of <('T, T2, T3, T4, T5)>)> Method (String, Func<(Of <('T, T2, T3, T4, T5, T)>)>, Object, String, CacheSettings)

Gets the multi map.

Namespace: SimpleEssentials.DataProvider
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#  
VB  
C++

```csharp
public IEnumerable<T> GetMultiMap<T, T2, T3, T4, T5>(
    string sql,
    Func<T, T2, T3, T4, T5, T> func,
    Object param,
    string splitOn,
    CacheSettings cacheSettings
)
```

```vbnet
Public Function GetMultiMap(Of T, T2, T3, T4, T5) ( _
    sql As String, _
    func As Func(Of T, T2, T3, T4, T5, T), _
    param As Object, _
    splitOn As String, _
    cacheSettings As CacheSettings _
) As IEnumerable(Of T)
```

```cpp
public:
    template<typename T, typename T2, typename T3, typename T4, typename T5>
    IEnumerable<T>^ GetMultiMap(
        String^ sql,
        Func<T, T2, T3, T4, T5, T>^ func,
        Object^ param,
        String^ splitOn,
        CacheSettings^ cacheSettings
    )
```

**Type Parameters**

T  
The type of the 2.

T2  
The type of the 2.

T3  
The type of the 3.

T4  
The type of the 4.
T5
The type of the 5.

Parameters

sql
Type: String
The SQL.

func
Type: Func<(Of <('T, T2, T3, T4, T5, T)>))
The function.

param
Type: Object
The parameter.

splitOn
Type: String
The split on.

cacheSettings
Type: SimpleEssentials.Cache...CacheSettings
The cache settings.

ReturnValue

IEnumerable<T>.
See Also

DbDataProvider Class
GetMultiMap Overload
SimpleEssentials.DataProvider Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
DbDataProvider..::..GetMultiMap<(O
<(<'T, T2, T3, T4, T5, T6>)>)>
Method (String, Func<(Of <(<'T, T2,
T3, T4, T5, T6, T>)>)>, Object,
String, CacheSettings)

Gets the multi map.

Namespace:  SimpleEssentials.DataProvider
Assembly:  SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#

```csharp
public IEnumerable<T> GetMultiMap<T, T2, T3, T4, T5, T6>(
    string sql,
    Func<T, T2, T3, T4, T5, T6, T> func,
    Object param,
    string splitOn,
    CacheSettings cacheSettings
)
```

VB

```vbnet
Public Function GetMultiMap(Of T, T2, T3, T4, T5, T6)(
    _
    sql As String, _
    func As Func(Of T, T2, T3, T4, T5, T6, T), _
    param As Object, _
    splitOn As String, _
    cacheSettings As CacheSettings _
) As IEnumerable(Of T)
```

C++

```cpp
public:
    template<typename T, typename T2, typename T3, typename T4, typename T5, typename T6>
    IEnumerable<T>^ GetMultiMap(
        String^ sql,
        Func<T, T2, T3, T4, T5, T6, T>^ func,
        Object^ param,
        String^ splitOn,
        CacheSettings^ cacheSettings
    )
```

Type Parameters

T
   The type of the 1.

T2
   The type of the 2.

T3
   The type of the 3.

T4
   The type of the 4.
The type of the 5.
The type of the 6.

**Parameters**

`sql`
Type: `String`
The SQL.

`func`
Type: `Func<(Of <(<'T, T2, T3, T4, T5, T6, T'>)>>)`
The function.

`param`
Type: `Object`
The parameter.

`splitOn`
Type: `String`
The split on.

`cacheSettings`
Type: `SimpleEssentials.Cache..::..CacheSettings`
The cache settings.

**Return Value**

`IEnumerable<T>`.
See Also

DbDataProvider Class  
GetMultiMap Overload  
SimpleEssentials.DataProvider Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
DbDataProvider..::..Insert<(Of <('T>')>) Method

Inserts a strongly typed object into a table.

Namespace: SimpleEssentials.DataProvider
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#

public bool Insert<T>(
    T data,
    CacheSettings cacheSettings
)

VB

Public Function Insert(Of T) ( _
    data As T, _
    cacheSettings As CacheSettings _
) As Boolean

C++

public: bool^ Insert(<typename T>
    bool^ Insert( 
    T data,
    CacheSettings^ cacheSettings
)

Type Parameters

T

Parameters

data
    Type: T
    The data.

cacheSettings
    Type: SimpleEssentials.Cache::..::CacheSettings
    The cache settings.

Return Value
True if successful, false if not
See Also

DbDataProvider Class
SimpleEssentials.DataProvider Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
DbDataProvider:::..InsertAndReturn Method
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>InsertAndReturnId&lt;Of &lt;&lt;'(T)&gt;&gt;&gt;(T, CacheSettings)</td>
<td>Inserts an object into a table and then return its ID.</td>
</tr>
<tr>
<td>InsertAndReturnId&lt;Of &lt;&lt;'(T)&gt;&gt;&gt;(String, T, CacheSettings)</td>
<td>Inserts an object into a table and then return its ID.</td>
</tr>
</tbody>
</table>
See Also

DbDataProvider Class
DbDataProvider Members
SimpleEssentials.DataProvider Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
DbData<br>Provider..::..InsertAndReturn<br>\(<\langle 'T' \rangle \rangle\) Method (\(T,\)<br>CacheSettings)<br><br>Inserts an object into a table and then return its ID.<br><br>**Namespace:** SimpleEssentials.DataProvider<br>**Assembly:** SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#

```csharp
public int InsertAndReturnId<T>(
    T data,
    CacheSettings cacheSettings
)
```

VB

```vbnet
Public Function InsertAndReturnId(Of T) ( _
    data As T, _
    cacheSettings As CacheSettings _
) As Integer
```

C++

```cpp
public:
    template<typename T>
    int^ InsertAndReturnId(
        T data,
        CacheSettings^ cacheSettings
    )
```

Type Parameters

T

Parameters

data
    Type: T
    The data.

cacheSettings
    Type: SimpleEssentials.Cache..::..CacheSettings
    The cache settings.

Return Value
ID of the newly inserted row
See Also

`DbDataProvider Class`
`InsertAndReturnId Overload`
`SimpleEssentials.DataProvider Namespace`

*Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.*
DbDataProvider..::..InsertAndReturn<?('T')> Method (String, T, CacheSettings)

Inserts an object into a table and then return its ID.

Namespace: SimpleEssentials.DataProvider
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#
VB
C++

public int InsertAndReturnId<T>(
    string sql,
    T data,
    CacheSettings cacheSettings
)

Public Function InsertAndReturnId(Of T) ( _
    sql As String, _
    data As T, _
    cacheSettings As CacheSettings _
) As Integer

public:
generic<typename T>
int^ InsertAndReturnId(    
    String^ sql,
    T data,
    CacheSettings^ cacheSettings
)

Type Parameters

T

Parameters

sql
    Type: String
    The SQL.

data
    Type: T
    The data.
cacheSettings
  Type: SimpleEssentials.Cache::CacheSettings
  The cache settings.

Return Value

ID of the newly inserted row
See Also

DbDataProvider Class
InsertAndReturnId Overload
SimpleEssentials.DataProvider Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
DbDataProvider:::..InsertList Method
### Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>InsertList(Of &lt;&lt;(T)&gt;&gt;)(IEnumerable(Of &lt;&lt;(T)&gt;&gt;), CacheSettings)</code></td>
<td>Insert a list of objects into a table.</td>
</tr>
<tr>
<td><code>InsertList(Of &lt;&lt;(T)&gt;&gt;)(IEnumerable(Of &lt;&lt;(T)&gt;&gt;), String, CacheSettings)</code></td>
<td>Insert a list of objects into a table.</td>
</tr>
</tbody>
</table>
See Also

DbDataProvider Class
DbDataProvider Members
SimpleEssentials.DataProvider Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
DbDataProvider::..InsertList<Of (Of (T))> Method
(IEnumerable<Of (T)>, CacheSettings)

Insert a list of objects into a table.

Namespace: SimpleEssentials.DataProvider
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#

```csharp
public int InsertList<T>(
    IEnumerable<T> data,
    CacheSettings cacheSettings
)
```

VB

```vbnet
Public Function InsertList(Of T) ( _
    data As IEnumerable(Of T), _
    cacheSettings As CacheSettings _
) As Integer
```

C++

```cpp
public:
    template<typename T> int
    ^ InsertList(
        IEnumerable<T>^ data,
        CacheSettings^ cacheSettings
    )
```

Type Parameters

T

Parameters

data
    Type: IEnumerable<(Of (<'T'>)>)
The data.

cacheSettings
    Type: SimpleEssentials.Cache..::.CacheSettings
    The cache settings.

Return Value
True if successful, false if not
See Also

DbDataProvider Class
InsertList Overload
SimpleEssentials.DataProvider Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
DbDataProvider..::..InsertList<(Of <('<T>')>)) Method
(Enumerable<(Of <('<T>')>),
String, CacheSettings)

Insert a list of objects into a table.

Namespace:  SimpleEssentials.DataProvider
Assembly:  SimpleEssentials (in SimpleEssentials.dll)
**Syntax**

**C#**
```csharp
public int InsertList<T>(
    IEnumerable<T> data,
    string sql,
    CacheSettings cacheSettings
)
```

**VB**
```vbnet
Public Function InsertList(Of T) ( _
    data As IEnumerable(Of T), _
    sql As String, _
    cacheSettings As CacheSettings _
) As Integer
```

```cpp
public:
generic<typename T>
int ^ InsertList(
    IEnumerable<T>^ data,
    String^ sql,
    CacheSettings^ cacheSettings
)
```

**Type Parameters**

**T**

**Parameters**

**data**

Type: **IEnumerable**<Of &<"T">>

The data.

**sql**

Type: **String**

The SQL.
cacheSettings
   Type: SimpleEssentials.Cache::CacheSettings
   The cache settings.

**Return Value**

Number of rows effected
See Also

DbDataProvider Class
InsertList Overload
SimpleEssentials.DataProvider Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
DbDataProvider..::.Update(Of T) Method

Updates te row in the table by the object passed.

Namespace: SimpleEssentials.DataProvider
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#

VB

C++

public bool Update<T>(
    T data,
    CacheSettings cacheSettings
)

Public Function Update(Of T) ( _
    data As T, _
    cacheSettings As CacheSettings _
) As Boolean

public
generic<typename T>
bool^ Update(
    bool^ Update(
        T data,
        CacheSettings^ cacheSettings
    )

Type Parameters

T

Parameters

data
    Type: T
    The data.

cacheSettings
    Type: SimpleEssentials.Cache..::.CacheSettings
    The cache settings.

Return Value
True if successful, false if not
See Also

DbData Provider Class
SimpleEssentials.Data Provider Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
IDataProvider Interface

Namespace: SimpleEssentials.DataProvider
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

public interface IDataProvider

Public Interface IDataProvider

public interface class IDataProvider
See Also

SimpleEssentials.DataProvider Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
IDataProvider Methods

The IDataProvider type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Equals(System.Object)</code></td>
<td>Determines whether the specified object is equal to the current object.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td><code>Finalize</code></td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td><code>GetHashCode</code></td>
<td>Serves as the default hash function.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td><code>GetType</code></td>
<td>Gets the <a href="#">Type</a> of the current instance.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td><code>MemberwiseClone</code></td>
<td>Creates a shallow copy of the current <a href="#">Object</a>.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td><code>ToString</code></td>
<td>Returns a string that represents the current object.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">Object</a>.)</td>
</tr>
</tbody>
</table>
See Also

IDataProvider Interface
SimpleEssentials.DataProvider Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
IDbProvider Interface

Database Provider Interface

Namespace: SimpleEssentials.DataProvider
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#
VB
C++

public interface IDbProvider : IDataProvider

Public Interface IDbProvider _
Inherits IDataProvider

public interface class IDbProvider : IDataProvider
See Also

IDbProvider Members
SimpleEssentials.DataProvider Namespace
SimpleEssentials.DataProvider..IDataProvider

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
**IDbProvider Members**

The [IDbProvider](#) type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BulkInsert&lt;Of &lt;&lt;'(T)&gt;&gt;()</td>
<td>TODO. Currently Not working</td>
</tr>
<tr>
<td>CreateTable&lt;Of &lt;&lt;'(T)&gt;&gt;()</td>
<td>Creates a table in the database.</td>
</tr>
<tr>
<td>Delete&lt;Of &lt;&lt;'(T)&gt;&gt;()</td>
<td>Deleted the row in the table.</td>
</tr>
<tr>
<td>Equals(System.Object)</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from <strong>Object</strong>.)</td>
</tr>
<tr>
<td>Execute</td>
<td>Executes the specified SQL.</td>
</tr>
<tr>
<td>ExecuteScalar</td>
<td>Executes the scalar.</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from <strong>Object</strong>.)</td>
</tr>
<tr>
<td>Get&lt;Of &lt;&lt;'(T)&gt;&gt;()&gt;(Object, CacheSettings)</td>
<td>Get a row from the database by ID.</td>
</tr>
<tr>
<td>Get&lt;Of &lt;&lt;'(T)&gt;&gt;()&gt;(CacheSettings)</td>
<td>Get all rows from a table.</td>
</tr>
<tr>
<td>Get&lt;Of &lt;&lt;'(T)&gt;&gt;()&gt;(Expression&lt;Of &lt;&lt;'(Func&lt;Of &lt;&lt;'(T, Boolean&gt;)&gt;&gt;()&gt;&gt;&gt;), CacheSettings)</td>
<td>Get all rows that match the expression given</td>
</tr>
<tr>
<td>GetByParameters&lt;Of &lt;&lt;'(T)&gt;&gt;()</td>
<td>Gets all rows by a sql statement and optional parameters.</td>
</tr>
<tr>
<td>GetByType&lt;Of &lt;&lt;'(T)&gt;&gt;()</td>
<td>Get all rows from a table.</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from <strong>Object</strong>.)</td>
</tr>
<tr>
<td>GetMultiMap&lt;Of &lt;&lt;'(T, T2)&gt;&gt;()</td>
<td>Builds an object that references multiple tables/rows.</td>
</tr>
<tr>
<td>GetMultiMap&lt;Of &lt;&lt;'(T, T2, T3)&gt;&gt;()</td>
<td>Gets the multi map.</td>
</tr>
</tbody>
</table>
Gets the multi map.

GetMultiMap<(Of <<'(T, T2, T3, T4, T5)>>)>(String, Func<(Of <<'(T, T2, T3, T4, T5, T)>>)>, Object, String, CacheSettings)

GetMultiMap<(Of <<'(T, T2, T3, T4, T5, T6)>>)>(String, Func<(Of <<'(T, T2, T3, T4, T5, T6, T)>>)>, Object, String, CacheSettings)

GetType

Insert<(Of <<'(T)>>)>

InsertAndReturnId<(Of <<'(T)>>)>(T, CacheSettings)

InsertAndReturnId<(Of <<'(T)>>)>(String, T, CacheSettings)

InsertList<(Of <<'(T)>>)>(IEnumerable<(Of <<'(T)>>)>, CacheSettings)

InsertList<(Of <<'(T)>>)>(IEnumerable<(Of <<'(T)>>)>, String, CacheSettings)

MemberwiseClone

ToString

Update<(Of <<'(T)>>)>

Obsolete.

Insert a list of objects into a table.

Obsolete.

Insert a list of objects into a table.

Inserts an object into a table and then return its ID.

Obsoletes.

Inserts an object into a table and then return its ID.

Inserts a strongly typed object into a table.

Inserts an object into a table and then return its ID.
See Also

IDbProvider Interface
SimpleEssentials.DataProvider Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
IDbProvider Methods

The IDbProvider type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>![BulkInsert](Of &lt;&lt;(T)&gt;&gt;)</td>
<td>TODO. Currently Not working</td>
</tr>
<tr>
<td>![CreateTable](Of &lt;&lt;(T)&gt;&gt;)</td>
<td>Creates a table in the database.</td>
</tr>
<tr>
<td>![Delete](Of &lt;&lt;(T)&gt;&gt;)</td>
<td>Deleted the row in the table.</td>
</tr>
<tr>
<td>![Equals(System.Object)]</td>
<td>Determines whether the specified object is equal to the current object.</td>
</tr>
<tr>
<td>![Execute](Of &lt;&lt;(T)&gt;&gt;)</td>
<td>(Inherited from Object.)</td>
</tr>
<tr>
<td>![ExecuteScalar]</td>
<td>Executes the specified SQL.</td>
</tr>
<tr>
<td>![Finalize]</td>
<td>Allows an object to try to free resources and perform other cleanup</td>
</tr>
<tr>
<td>![Get](Of &lt;&lt;(T)&gt;&gt;)</td>
<td>operations before it is reclaimed by garbage collection.</td>
</tr>
<tr>
<td>![Get](Object, CacheSettings)</td>
<td>(Inherited from Object.)</td>
</tr>
<tr>
<td><img src="CacheSettings" alt="Get" /></td>
<td>Get a row from the database by ID.</td>
</tr>
<tr>
<td>![Get](Expression&lt;Of &lt;&lt;(Func&lt;Of &lt;&lt;(T, Boolean)&gt;&gt;&gt;)&gt;&gt;&gt;, CacheSettings)</td>
<td>Get all rows that match the expression given</td>
</tr>
<tr>
<td>![GetByParameters](Of &lt;&lt;(T)&gt;&gt;)</td>
<td>Gets all rows by a sql statement and optional parameters.</td>
</tr>
<tr>
<td>![GetByType](Of &lt;&lt;(T)&gt;&gt;)</td>
<td>Get all rows from a table.</td>
</tr>
<tr>
<td>![GetHashCode]</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>![GetMultiMap](Of &lt;&lt;(T, T2)&gt;&gt;)</td>
<td>Builds an object that references multiple tables/rows.</td>
</tr>
<tr>
<td>![GetMultiMap](String, Func&lt;Of &lt;&lt;(T, T2, T)&gt;&gt;&gt;, Object, String, CacheSettings)</td>
<td>Gets the multi map.</td>
</tr>
</tbody>
</table>
Gets the multi map.

gets the multi map.

Gets the multi map.

Gets the **Type** of the current instance.
(Inherited from **Object**.)

Inserts an object into a table and then return its ID.

**Obsolete.**
Inserts an object into a table and then return its ID.

Insert a list of objects into a table.

**Obsolete.**
Insert a list of objects into a table.

Creates a shallow copy of the current **Object**.
(Inherited from **Object**.)

Returns a string that represents the current object.
(Inherited from **Object**.)

Updates the row in the table by the object passed.
See Also

IDbProvider Interface
SimpleEssentials.DataProvider Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
IDbProvider...:::BulkInsert<(Of <'T'>)> Method

TODO. Currently Not working

Namespace: SimpleEssentials.DataProvider
Assembly: SimpleEssentials (in SimpleEssentials.dll)
## Syntax

**C#**

```csharp
void BulkInsert<T>(
    IEnumerable<T> data,
    string tableName,
    CacheSettings cacheSettings
)
```

**VB**

```vbnet
Sub BulkInsert(Of T) (
    data As IEnumerable(Of T), _
    tableName As String, _
    cacheSettings As CacheSettings _
)
```

**C++**

```cpp
generic<typename T>
void BulkInsert(
    IEnumerable<T>^ data,
    String^ tableName,
    CacheSettings^ cacheSettings
)
```

### Type Parameters

**T**

### Parameters

- **data**
  - Type: `IEnumerable<T>`
  - The data.

- **tableName**
  - Type: `String`
  - Name of the table.

- **cacheSettings**
Type: SimpleEssentials.Cache...CacheSettings
The cache settings.
See Also

IDbProvider Interface
SimpleEssentials.DataProvider Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
IDbProvider::CreateTable(Of <T>) Method

Creates a table in the database.

Namespace: SimpleEssentials.DataProvider
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#
VB
C++

`bool CreateTable<T>()`

Function `CreateTable(Of T)` As `Boolean`

generic<typename T>
`bool^ CreateTable()`

**Type Parameters**

T

**Return Value**

True is successfull, false if table exists already.
See Also

IDbProvider Interface
SimpleEssentials.DataProvider Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
**IDbProvider..::..Delete<(Of <('T')>)> Method**

Deleted the row in the table.

**Namespace:** SimpleEssentials.DataProvider  
**Assembly:** SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#

```csharp
bool Delete<T>(
    T data,
    CacheSettings cacheSettings,
    string fieldKey
)
```

VB

```vbnet
Function Delete(Of T) ( _
    data As T, _
    cacheSettings As CacheSettings, _
    fieldKey As String _
) As Boolean
```

c++

```cpp
generic<typename T>
bool^ Delete(
    T data,
    CacheSettings^ cacheSettings,
    String^ fieldKey
)
```

Type Parameters

T

Parameters

data
  Type: T
  The data.

cacheSettings
  Type: SimpleEssentials.Cache..::.CacheSettings
  The cache settings.

fieldKey
Type:  String
The field key.

Return Value

True if successful, false if not
See Also

**IDbProvider Interface**  
**SimpleEssentials.DataProvider Namespace**

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
IDbProvider...:::Execute Method

Executes the specified SQL.

Namespace: SimpleEssentials.DataProvider
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#  
VB  
C++

```csharp
int Execute(
    string sql,
    Object param,
    CacheSettings cacheSettings,
    bool invalidateCache
)
```

Function Execute (
    sql As String, _
    param As Object, _
    cacheSettings As CacheSettings, _
    invalidateCache As Boolean _
) As Integer

```csharp
int^ Execute(
    String^ sql,
    Object^ param,
    CacheSettings^ cacheSettings,
    bool^ invalidateCache
)
```

Parameters

sql
  Type: String
  The SQL.

param
  Type: Object
  The parameter.

cacheSettings
  Type: SimpleEssentials.Cache..::..CacheSettings
  The cache settings.
invalidateCache
   Type: **Boolean**
   if set to true [invalidate cache].

**Return Value**

TODO
See Also

IDbProvider Interface
SimpleEssentials.DataProvider Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
IDbProvider..::..ExecuteScalar Method

Executes the scalar.

**Namespace:** SimpleEssentials.DataProvider  
**Assembly:** SimpleEssentials (in SimpleEssentials.dll)
Syntax

**C#**

```csharp
int ExecuteScalar(
    string sql,
    Object param,
    CacheSettings cacheSettings,
    bool invalidateCache
)
```

**Function ExecuteScalar ( _
  sql As String, _
  param As Object, _
  cacheSettings As CacheSettings, _
  invalidateCache As Boolean _
) As Integer**

```csharp
int^ ExecuteScalar(
    String^ sql,
    Object^ param,
    CacheSettings^ cacheSettings,
    bool^ invalidateCache
)
```

**Parameters**

**sql**
- Type: *String*
- The SQL.

**param**
- Type: *Object*
- The parameter.

**cacheSettings**
- Type: *SimpleEssentials.Cache..::..CacheSettings*
- The cache settings.
invalidateCache
   Type: Boolean
   if set to true [invalidate cache].

Return Value

TODO
See Also

IDbProvider Interface
SimpleEssentials.DataProvider Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
IDbProvider...Get Method
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Get&lt;(Of &lt;&lt;'(T)&gt;&gt;)(Object, CacheSettings)</code></td>
<td>Get a row from the database by ID.</td>
</tr>
<tr>
<td><code>Get&lt;(Of &lt;&lt;'(T)&gt;&gt;)(CacheSettings)</code></td>
<td>Get all rows from a table.</td>
</tr>
<tr>
<td><code>Get&lt;(Of &lt;&lt;'(T)&gt;&gt;)(Expression&lt;(Of &lt;&lt;'(Func&lt;(Of &lt;&lt;'(T, Boolean)&gt;&gt;)&gt;&gt;, CacheSettings))</code></td>
<td>Get all rows that match the expression given</td>
</tr>
</tbody>
</table>
See Also

IDbProvider Interface
IDbProvider Members
SimpleEssentials.DataProvider Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
IDbProvider..::..Get<(Of <('T')>)>
Method (Object, CacheSettings)

Get a row from the database by ID.

Namespace:  SimpleEssentials.DataProvider
Assembly:  SimpleEssentials (in SimpleEssentials.dll)
C#
Function Get<T>(
    T id,
    CacheSettings cacheSettings
) As T

Function Get(Of T) ( _
    id As Object, _
    cacheSettings As CacheSettings _
) As T

generic<typename T>
T Get ( 
    Object id,
    CacheSettings cacheSettings
)

Type Parameters

T

Parameters

id
    Type: Object
    The identifier.

cacheSettings
    Type: SimpleEssentials.Cache...CacheSettings
    The cache settings.

Return Value

Strongly typed object
See Also

IDbProvider Interface
Get Overload
SimpleEssentials.DataProvider Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
IDbProvider:::Get<(Of <('T')>)>
Method (CacheSettings)

Get all rows from a table.

Namespace: SimpleEssentials.DataProvider
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#  
VB  
C++

```csharp
IEnumerable<T> Get<T>(
    CacheSettings cacheSettings
)
```

```vbnet
Function Get(Of T) ( _
    cacheSettings As CacheSettings _
) As IEnumerable(Of T)
```

generic<typename T>

```csharp
IEnumerable<T>^ Get(
    CacheSettings^ cacheSettings
)
```

Type Parameters

T

Parameters

cacheSettings
    Type: SimpleEssentials.Cache..::.CacheSettings
    The cache settings.

Return Value

A list of strongly typed objects
See Also

IDbProvider Interface
Get Overload
SimpleEssentials.DataProvider Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
Get all rows that match the expression given

**Namespace:**  [SimpleEssentials.DataProvider](#)  
**Assembly:**  SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#

`IEnumerable<T> Get<T>(
    Expression<Func<T, bool>> expression,
    CacheSettings cacheSettings
)
`

Function Get(Of T) ( _
  expression As Expression(Of Func(Of T, Boolean)), _
  cacheSettings As CacheSettings _
) As IEnumerable(Of T)

generic<typename T>

`IEnumerable<T>^ Get(
    Expression<Func<T, bool>^>^ expression,
    CacheSettings^ cacheSettings
)
`

Type Parameters

T

Parameters

expression
  Type: `Expression<(Of (<'Func<(Of (<'T, Boolean)>)>>)>)>
  The expression.

cacheSettings
  Type: `SimpleEssentials.Cache..::.CacheSettings
  The cache settings.

Return Value

Gets a list of strongly typed objects
See Also

IDbProvider Interface
Get Overload
SimpleEssentials.DataProvider Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
IDbProvider::GetByParameters<
(("T"))> Method

Gets all rows by a sql statement and optional parameters.

Namespace:  SimpleEssentials.DataProvider
Assembly:  SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#
VB
C++

**IEnumerable**<T> GetByParameters<T>(
    string sql,
    Object param,
    CacheSettings cacheSettings
)

Function GetByParameters(Of T) (   
    sql As String, _
    param As Object, _
    cacheSettings As CacheSettings _
) As IEnumerable(Of T)

generic<typename T>  
IEnumerable<T> GetByParameters(
    String^ sql,
    Object^ param,
    CacheSettings^ cacheSettings
)

**Type Parameters**

T

**Parameters**

sql

  Type: **String**
  The SQL.

param

  Type: **Object**
  The parameter.

cacheSettings
Type: `SimpleEssentials.Cache...CacheSettings`

The cache settings.

**Return Value**
See Also

IDbProvider Interface
SimpleEssentials.DataProvider Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
IDbProvider:::GetByType<Of <('T')?>()> Method

Get all rows from a table.

Namespace: SimpleEssentials.DataProvider
Assembly: SimpleEssentials (in SimpleEssentials.dll)
**Syntax**

C#  
GetByType<T>(
    CacheSettings cacheSettings
)

Function GetByType(Of T) ( _
    cacheSettings As CacheSettings _
) As IEnumerable(Of T)

generic<typename T>
IEnumerable<T> GetByType(
    CacheSettings^ cacheSettings
)

**Type Parameters**

T

**Parameters**

cacheSettings
    Type: SimpleEssentials.Cache..::.CacheSettings
    The cache settings.

**Return Value**

List of strongly typed objects
See Also

**IDbProvider Interface**
**SimpleEssentials.DataProvider Namespace**

*Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.*
IDbProvider...::..GetMultiMap<(Of <('T, T2)>)> Method (String, Func<(Of <('T, T2, T)>)>), Object, String, CacheSettings)

Builds an object that references multiple tables/rows.

Namespace: SimpleEssentials.DataProvider
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#

```csharp
IEnumerable<T> GetMultiMap<T, T2>(
    string sql,
    Func<T, T2, T> func,
    Object param,
    string splitOn,
    CacheSettings cacheSettings
)
```

Function GetMultiMap(Of T, T2)(
    sql As String, _
    func As Func(Of T, T2, T), _
    param As Object, _
    splitOn As String, _
    cacheSettings As CacheSettings _
) As IEnumerable(Of T)

```csharp
IEnumerable<T>^ GetMultiMap(
    String^ sql, ^
    Func<T, T2, T>^ func, ^
    Object^ param, ^
    String^ splitOn, ^
    CacheSettings^ cacheSettings ^
)
```

Type Parameters

T
T2

The type of the 2.

Parameters

sql
Type: String
The SQL.

func
    Type: Func<(Of <$> T, T2, T)>>
    The function.

param
    Type: Object
    The parameter.

splitOn
    Type: String
    The split on.

cacheSettings
    Type: SimpleEssentials.Cache::CacheSettings
    The cache settings.

**Return Value**

Strongly typed object whos data is mapped for you
See Also

IDbProvider Interface
GetMultiMap Overload
SimpleEssentials.DataProvider Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
GetMultiMap<Of <('T, T2, T3>>>> Method (String, Func<Of <('T, T2, T3, T>>>>, Object, String, CacheSettings)

Gets the multi map.

Namespace: SimpleEssentials.DataProvider
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

**C#**

```csharp
IEnumerable<T> GetMultiMap<T, T2, T3>(
    string sql,
    Func<T, T2, T3, T> func,
    Object param,
    string splitOn,
    CacheSettings cacheSettings
)
```

**VB**

```vbnet
Function GetMultiMap(Of T, T2, T3) ( _
    sql As String, _
    func As Func(Of T, T2, T3, T), _
    param As Object, _
    splitOn As String, _
    cacheSettings As CacheSettings _
) As IEnumerable(Of T)
```

generic<typename T, typename T2, typename T3>

```csharp
IEnumerable<T>^ GetMultiMap(
    String^ sql,
    Func<T, T2, T3, T>^ func,
    Object^ param,
    String^ splitOn,
    CacheSettings^ cacheSettings
)
```

**Type Parameters**

T

The type of the 1.

T2

The type of the 2.

T3

The type of the 3.

**Parameters**
sql
    Type: String
    The SQL.

func
    Type: Func<(Of <('T, T2, T3, T)>)>
    The function.

param
    Type: Object
    The parameter.

splitOn
    Type: String
    The split on.

cacheSettings
    Type: SimpleEssentials.Cache..::.CacheSettings
    The cache settings.

Return Value
See Also

IDbProvider Interface
GetMultiMap Overload
SimpleEssentials.DataProvider Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
IDbProvider...:..GetMultiMap<(Of <('T, T2, T3, T4)>)> Method
(String, Func<(Of <('T, T2, T3, T4, T)>)>), Object, String,
CacheSettings)

Gets the multi map.

Namespace: SimpleEssentials.DataProvider
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#

```csharp
IEnumerable<T> GetMultiMap<T, T2, T3, T4>(
    string sql,
    Func<T, T2, T3, T4, T> func,
    Object param,
    string splitOn,
    CacheSettings cacheSettings
)
```

`IEnumerable<T>^ GetMultiMap<T, T2, T3, T4>(
    String^ sql,
    Func<T, T2, T3, T4, T>^ func,
    Object^ param,
    String^ splitOn,
    CacheSettings^ cacheSettings
)`

VB

```vbnet
Function GetMultiMap(Of T, T2, T3, T4) (_
    sql As String, _
    func As Func(Of T, T2, T3, T4, T), _
    param As Object, _
    splitOn As String, _
    cacheSettings As CacheSettings _
) As IEnumerable(Of T)
```

```vbnet
generic<typeName T, typeName T2, typeName T3, typeName T4>
IEnumerable<T>^ GetMultiMap(
    String^ sql,
    Func<T, T2, T3, T4, T>^ func,
    Object^ param,
    String^ splitOn,
    CacheSettings^ cacheSettings
)
```

C++

```cpp
std::vector<T> GetMultiMap(T, T2, T3, T4)(
    char* sql,
    Func<T, T2, T3, T4, T> func,
    Object param,
    char* splitOn,
    CacheSettings cacheSettings
)
```

`std::vector<T>^ GetMultiMap<T, T2, T3, T4>(
    char^ sql,
    Func<T, T2, T3, T4, T>^ func,
    Object^ param,
    char^ splitOn,
    CacheSettings^ cacheSettings
)`

Type Parameters

T
T2
The type of the 2.

T3
The type of the 3.

T4
The type of the 4.
**Parameters**

- **sql**
  - Type: `String`
  - The SQL.

- **func**
  - Type: `Func<(Of <('T, T2, T3, T4, T)>)>`
  - The function.

- **param**
  - Type: `Object`
  - The parameter.

- **splitOn**
  - Type: `String`
  - The split on.

- **cacheSettings**
  - Type: `SimpleEssentials.Cache::CacheSettings`
  - The cache settings.

**Return Value**
See Also

**IDbProvider Interface**
**GetMultiMap Overload**
**SimpleEssentials.DataProvider Namespace**

[Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.](#)
IDbProvider...:::GetMultiMap<(Of <('T, T2, T3, T4, T5)>)> Method (String, Func<Of <('T, T2, T3, T4, T5, T)>>, Object, String, CacheSettings)

Gets the multi map.

**Namespace:** SimpleEssentials.DataProvider
**Assembly:** SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#
VB
C++

```csharp
IEnumerable<T> GetMultiMap<T, T2, T3, T4, T5>(
    string sql,
    Func<T, T2, T3, T4, T5, T> func,
    Object param,
    string splitOn,
    CacheSettings cacheSettings
)
```

Function GetMultiMap(Of T, T2, T3, T4, T5) ( _
    sql As String, _
    func As Func(Of T, T2, T3, T4, T5, T), _
    param As Object, _
    splitOn As String, _
    cacheSettings As CacheSettings _
) As IEnumerable(Of T)

generic<typeName T, typeName T2, typeName T3, typeName T4, typeName T5>
IEnumerable<T>^ GetMultiMap(
    String^ sql,
    Func<T, T2, T3, T4, T5, T>^ func,
    Object^ param,
    String^ splitOn,
    CacheSettings^ cacheSettings
)
```

Type Parameters

T
T2
    The type of the 2.
T3
    The type of the 3.
T4
    The type of the 4.
T5
The type of the 5.

**Parameters**

**sql**
Type: **String**
The SQL.

**func**
Type: Func<(Of <('T, T2, T3, T4, T5, T)>)>The function.

**param**
Type: **Object**
The parameter.

**splitOn**
Type: **String**
The split on.

**cacheSettings**
Type: **SimpleEssentials.Cache...CacheSettings**
The cache settings.

**Return Value**
See Also

IDbProvider Interface
GetMultiMap Overload
SimpleEssentials.DataProvider Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
**IDbProvider...:::GetMultiMap(Of <('T, T2, T3, T4, T5, T6)>)>**

**Method** (String, Func(Of <('T, T2, T3, T4, T5, T6, T)>), Object, String, CacheSettings)

Gets the multi map.

**Namespace:** SimpleEssentials.DataProvider  
**Assembly:** SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#

```csharp
IEnumerable<T> GetMultiMap<T, T2, T3, T4, T5, T6>(
    string sql,
    Func<T, T2, T3, T4, T5, T6, T> func,
    Object param,
    string splitOn,
    CacheSettings cacheSettings
)
```

Function GetMultiMap(Of T, T2, T3, T4, T5, T6) ( _
    sql As String, _
    func As Func(Of T, T2, T3, T4, T5, T6, T), _
    param As Object, _
    splitOn As String, _
    cacheSettings As CacheSettings _
) As IEnumerable(Of T)

generic<typename T, typename T2, typename T3, typename T4, typename T5>
IEnumerable<T> GetMultiMap(
    String sql,
    Func<T, T2, T3, T4, T5, T6, T> func,
    Object param,
    String splitOn,
    CacheSettings cacheSettings
)

Type Parameters

T
T2
    The type of the 2.
T3
    The type of the 3.
T4
    The type of the 4.
T5
    The type of the 5.
The type of the 5.
T6
The type of the 6.

**Parameters**

sql
Type: **String**
The SQL.

func
Type: Func<(Of `<T, T2, T3, T4, T5, T6, T>`)>
The function.

param
Type: **Object**
The parameter.

splitOn
Type: **String**
The split on.

cacheSettings
Type: **SimpleEssentials.Cache..CacheSettings**
The cache settings.

**Return Value**
See Also

IDbProvider Interface
GetMultiMap Overload
SimpleEssentials.DataProvider Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
IDbProvider:::..Insert<T> Method

Inserts a strongly typed object into a table.

Namespace: SimpleEssentials.DataProvider
Assembly: SimpleEssentials (in SimpleEssentials.dll)
## Syntax

### C#
```csharp
bool Insert<T>(
    T data,
    CacheSettings cacheSettings
)
```

### VB
```vbnet
Function Insert(Of T) (_
    data As T, _
    cacheSettings As CacheSettings _
) As Boolean
```

### C++
```cpp
generic<typename T>
bool^ Insert(
    T data,
    CacheSettings^ cacheSettings
)
```

### Type Parameters

- **T**

### Parameters

- **data**
  - Type: T
  - The data.

- **cacheSettings**
  - Type: `SimpleEssentials.Cache..::.CacheSettings`
  - The cache settings.

### Return Value

True if successful, false if not
See Also

- IDbProvider Interface
- SimpleEssentials.DataProvider Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
IDbProvider::InsertAndReturnId Method
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>InsertAndReturnId&lt;(Of &lt;&lt;'(T)&gt;&gt;)(T, CacheSettings)</code></td>
<td>Inserts an object into a table and then return its ID.</td>
</tr>
<tr>
<td><code>InsertAndReturnId&lt;(Of &lt;&lt;'(T)&gt;&gt;)</code>(String, T, CacheSettings)`</td>
<td><strong>Obsolete.</strong> Inserts an object into a table and then return its ID.</td>
</tr>
</tbody>
</table>
See Also

IDbProvider Interface
IDbProvider Members
SimpleEssentials.DataProvider Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
IDbProvider:::..InsertAndReturnId<(<T>),> Method (T, CacheSettings)

Inserts an object into a table and then return its ID.

Namespace: SimpleEssentials.DataProvider
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#

```csharp
int InsertAndReturnId<T>(
    T data,
    CacheSettings cacheSettings
)
```

VB

```vbnet
Function InsertAndReturnId(Of T) ( _
    data As T, _
    cacheSettings As CacheSettings _
) As Integer

generic<typename T>
```n

```csharp
int^ InsertAndReturnId(
    T data,
    CacheSettings^ cacheSettings
)
```

Type Parameters

T

Parameters

data
    Type: T
    The data.

cacheSettings
    Type: SimpleEssentials.Cache..::.CacheSettings
    The cache settings.

Return Value

ID of the newly inserted row
See Also

IDbProvider Interface
InsertAndReturnId Overload
SimpleEssentials.DataProvider Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
IDbProvider::..InsertAndReturnId<(<'T'>)> Method (String, T, CacheSettings)

Inserts an object into a table and then return its ID.

Namespace: SimpleEssentials.DataProvider
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#
VB
C++

[ObsoleteAttribute]
int InsertAndReturnId<T>(
    string sql,
    T data,
    CacheSettings cacheSettings
)

<ObsoleteAttribute> _
Function InsertAndReturnId(Of T) ( _
    sql As String, _
    data As T, _
    cacheSettings As CacheSettings _
) As Integer

[ObsoleteAttribute]
generic<typename T>
int^ InsertAndReturnId(
    String^ sql,
    T data,
    CacheSettings^ cacheSettings
)

Type Parameters

T

Parameters

sql
    Type: String
    The SQL.

data
    Type: T
The data.

cacheSettings
   Type: SimpleEssentials.Cache::CacheSettings
   The cache settings.

Return Value

ID of the newly inserted row
See Also

IDbProvider Interface
InsertAndReturnId Overload
SimpleEssentials.DataProvider Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
IDbProvider::InsertList Method
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>InsertList&lt;Of &lt;&lt;(T)&gt;&gt;&gt;(IEnumerable&lt;Of &lt;&lt;(T)&gt;&gt;, CacheSettings)</td>
<td>Insert a list of objects into a table. <strong>Obsolete.</strong> Insert a list of objects into a table.</td>
</tr>
</tbody>
</table>
See Also

IDbProvider Interface
IDbProvider Members
SimpleEssentials.DataProvider Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
IDbProvider...::..InsertList<(Of <("'T"">)> )> Method (IEnumerable<(Of <("'T"">)> )>, CacheSettings)

Insert a list of objects into a table.

Namespace: SimpleEssentials.DataProvider
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

**C#**

```csharp
int InsertList<T>(
    IEnumerable<T> data,
    CacheSettings cacheSettings
)
```

**VB**

```vbnet
Function InsertList(Of T) ( _
    data As IEnumerable(Of T), _
    cacheSettings As CacheSettings _
) As Integer
```

generic<typename T>

```csharp
int^ InsertList(
    IEnumerable<T>^ data,
    CacheSettings^ cacheSettings
)
```

**Type Parameters**

T

**Parameters**

data  
Type: `IEnumerable<Of (Of (T)>)`  
The data.

cacheSettings  
Type: `SimpleEssentials.Cache..::.CacheSettings`  
The cache settings.

**Return Value**

True if successful, false if not
See Also

IDbProvider Interface
InsertList Overload
SimpleEssentials.DataProvider Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
IDbProvider..::..InsertList<(Of <('T')>)> Method
(IEnumerable<(Of <('T')>)>, String, CacheSettings)

Insert a list of objects into a table.

Namespace: SimpleEssentials.DataProvider
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#

```csharp
[ObsoleteAttribute]
int InsertList<T>(
    IEnumerable<T> data,
    string sql,
    CacheSettings cacheSettings
)
```

```csharp
<ObsoleteAttribute> _
Function InsertList(Of T) ( _
    data As IEnumerable(Of T), _
    sql As String, _
    cacheSettings As CacheSettings _
) As Integer
```

```csharp
[ObsoleteAttribute]
generic<typename T>
int^ InsertList(
    IEnumerable<T>^ data,
    String^ sql,
    CacheSettings^ cacheSettings
)
```

Type Parameters

T

Parameters

data
  Type: IEnumerable(Of `(T)`)
  The data.

sql
  Type: String
The SQL.

cacheSettings
   Type: SimpleEssentials.Cache..CacheSettings
   The cache settings.

Return Value

Number of rows effected
See Also

IDbProvider Interface
InsertList Overload
SimpleEssentials.DataProvider Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
IDbProvider...::..Update<(Of ((<'T'>))>) Method

Updates the row in the table by the object passed.

Namespace: SimpleEssentials.DataProvider
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#

VB

C++

```csharp
bool Update<T>(
    T data,
    CacheSettings cacheSettings
)
```

```vbnet
Function Update(Of T) ( _
    data As T, _
    cacheSettings As CacheSettings _
) As Boolean
```

```cpp
generic<typename T>
bool^ Update(
    T data,
    CacheSettings^ cacheSettings
)
```

**Type Parameters**

T

**Parameters**

data

Type: T
The data.

cacheSettings

Type: SimpleEssentials.Cache...CacheSettings
The cache settings.

**Return Value**

True if successful, false if not
See Also

IDbProvider Interface
SimpleEssentials.DataProvider Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
IFileProvider Interface

Namespace:  SimpleEssentials.DataProvider
Assembly:  SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#
VB
C++

public interface IFileProvider : IDataProvider

Public Interface IFileProvider _
Inherits IDataProvider

public interface class IFFileProvider : IDataProvider
See Also

SimpleEssentials.DataProvider Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
IFileProvider Methods

The IFileProvider type exposes the following members.
### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Equals(System.Object)</code></td>
<td>Determines whether the specified object is equal to the current object. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><code>Finalize</code></td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><code>GetHashCode</code></td>
<td>Serves as the default hash function. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><code>GetType</code></td>
<td>Gets the <code>Type</code> of the current instance. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><code>MemberwiseClone</code></td>
<td>Creates a shallow copy of the current <code>Object</code>. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><code>ToString</code></td>
<td>Returns a string that represents the current object. (Inherited from <code>Object</code>.)</td>
</tr>
</tbody>
</table>
See Also

IFileProvider Interface
SimpleEssentials.DataProvider Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
SimpleEssentials.DataStore
Namespace
# Classes

<table>
<thead>
<tr>
<th>Class</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>🌱 DbStore</td>
<td></td>
</tr>
</tbody>
</table>
Interfaces

<table>
<thead>
<tr>
<th>Interface</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IDataStore</td>
<td></td>
</tr>
<tr>
<td>IDbStore</td>
<td></td>
</tr>
</tbody>
</table>

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
DbStore Class

Namespace: SimpleEssentials.DataStore
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#  

public class DbStore : IDataStore

VB

Public Class DbStore  
    Implements IDataStore

C++

public ref class DbStore : IDataStore
Inheritance Hierarchy

Object
SimpleEssentials.DataStore...DbStore
See Also

DbStore Members
SimpleEssentials.DataStore Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
DbStore Members

The DbStore type exposes the following members.
## Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DbStore</td>
<td></td>
</tr>
</tbody>
</table>
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add&lt;(Of &lt;&lt;(T)&gt;&gt;)</td>
<td></td>
</tr>
<tr>
<td>AddAndReturnId&lt;(Of &lt;&lt;(T)&gt;&gt;)</td>
<td></td>
</tr>
<tr>
<td>AddList&lt;(Of &lt;&lt;(T)&gt;&gt;)</td>
<td></td>
</tr>
<tr>
<td>BulkInsert&lt;(Of &lt;&lt;(T)&gt;&gt;)</td>
<td></td>
</tr>
<tr>
<td>Delete&lt;(Of &lt;&lt;(T)&gt;&gt;)</td>
<td></td>
</tr>
<tr>
<td>Equals(System.Object)</td>
<td>Determines whether the specified object is equal to the current object.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <strong>Object</strong>)</td>
</tr>
<tr>
<td>Execute</td>
<td></td>
</tr>
<tr>
<td>ExecuteScalar</td>
<td></td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup</td>
</tr>
<tr>
<td></td>
<td>operations before it is reclaimed by garbage collection.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <strong>Object</strong>)</td>
</tr>
<tr>
<td>Get&lt;(Of &lt;&lt;(T)&gt;&gt;)</td>
<td></td>
</tr>
<tr>
<td>GetByParameters&lt;(Of &lt;&lt;(T)&gt;)</td>
<td></td>
</tr>
<tr>
<td>GetByType&lt;(Of &lt;&lt;(T)&gt;)</td>
<td></td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <strong>Object</strong>)</td>
</tr>
<tr>
<td>GetMultiMap&lt;(Of &lt;&lt;(T, T2)&gt;)</td>
<td></td>
</tr>
<tr>
<td>(String, Func&lt;(Of &lt;&lt;(T, T2, T)&gt;&gt;)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(Object, String)</td>
</tr>
<tr>
<td>GetMultiMap&lt;(Of &lt;&lt;(T, T2, T3,)&gt;)</td>
<td></td>
</tr>
<tr>
<td>(String, Func&lt;(Of &lt;&lt;(T, T2, T3, T,)&gt;)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(Object, String)</td>
</tr>
<tr>
<td>GetMultiMap&lt;(Of &lt;&lt;(T, T2, T3, T4,)&gt;)</td>
<td></td>
</tr>
<tr>
<td>(String, Func&lt;(Of &lt;&lt;(T, T2, T3, T4, T,)&gt;)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(Object, String)</td>
</tr>
<tr>
<td>GetMultiMap&lt;(Of &lt;&lt;(T, T2, T3, T4,)&gt;)</td>
<td></td>
</tr>
</tbody>
</table>
T5>>>(String, Func<Of <<'(T, T2, T3, T4, T5, T)>>, Object, String)
GetMultiMap<Of <<'(T, T2, T3, T4, T5, T6)>>>(String, Func<Of <<'(T, T2, T3, T4, T5, T6, T)>>, Object, String)

GetType

MemberwiseClone

ToString

Update<Of <<'(T)>>>

Gets the Type of the current instance. (Inherited from Object.)
Creates a shallow copy of the current Object. (Inherited from Object.)
Returns a string that represents the current object. (Inherited from Object.)
See Also

DbStore Class
SimpleEssentials.DataStore Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
DbStore Constructor

Namespace: SimpleEssentials.DataStore
Assembly: SimpleEssentials (in SimpleEssentials.dll)
**Syntax**

**C#**

```csharp
public DbStore(string connectionString)
```

**VB**

```vbnet
Public Sub New( _
    connectionString As String _
)
```

**C++**

```c++
public: 
DbStore( 
    String^ connectionString
)
```

**Parameters**

connectionString  
Type: `String`
See Also

DbStore Class
SimpleEssentials.DataStore Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
DbStore Methods

The DbStore type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add&lt;(Of &lt;&lt;'(T)&gt;)&gt;</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td>AddAndReturnId&lt;(Of &lt;&lt;'(T)&gt;)&gt;</td>
<td></td>
</tr>
<tr>
<td>AddList&lt;(Of &lt;&lt;'(T)&gt;)&gt;</td>
<td></td>
</tr>
<tr>
<td>BulkInsert&lt;(Of &lt;&lt;'(T)&gt;)&gt;</td>
<td></td>
</tr>
<tr>
<td>Delete&lt;(Of &lt;&lt;'(T)&gt;)&gt;</td>
<td></td>
</tr>
<tr>
<td>Equals(System.Object)</td>
<td></td>
</tr>
<tr>
<td>Execute</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td>ExecuteScalar</td>
<td></td>
</tr>
<tr>
<td>Finalize</td>
<td></td>
</tr>
<tr>
<td>Get&lt;(Of &lt;&lt;'(T)&gt;)&gt;</td>
<td>Serves as the default hash function. (Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td>GetByParameters&lt;(Of &lt;&lt;'(T)&gt;)&gt;</td>
<td></td>
</tr>
<tr>
<td>GetByType&lt;(Of &lt;&lt;'(T)&gt;)&gt;</td>
<td></td>
</tr>
<tr>
<td>GetHashCode</td>
<td></td>
</tr>
<tr>
<td>GetMultiMap&lt;(Of &lt;&lt;'(T, T2)&gt;)&gt;</td>
<td></td>
</tr>
<tr>
<td>GetMultiMap&lt;(Of &lt;&lt;'(T, T2, T)&gt;)&gt;</td>
<td></td>
</tr>
<tr>
<td>GetMultiMap&lt;(Of &lt;&lt;'(T, T2, T3)&gt;)&gt;</td>
<td></td>
</tr>
<tr>
<td>GetMultiMap&lt;(Of &lt;&lt;'(T, T2, T3, T4)&gt;)&gt;</td>
<td></td>
</tr>
<tr>
<td>GetMultiMap&lt;(Of &lt;&lt;'(T, T2, T3, T4)&gt;)&gt;</td>
<td></td>
</tr>
</tbody>
</table>

[Object](#)
GetMultiMap<(String, Func<(Of <<'(T, T2, T3, T4, T5, T6>>)>, Object, String))

GetMultiMap<(String, Func<(Of <<'(T, T2, T3, T4, T5, T6>>)>, Object, String))

GetType

GetType

MemberwiseClone

MemberwiseClone

ToString

ToString

Update<(Of <<'(T)>>)>

Update<(Of <<'(T)>>)>

Gets the **Type** of the current instance.
(Inherited from **Object**.)

Creates a shallow copy of the current **Object**.
(Inherited from **Object**.)

Returns a string that represents the current object.
(Inherited from **Object**.)
See Also

DbStore Class
SimpleEssentials.DataStore Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
DbStore..::..Add<(Of <('T')>)>

Method

Namespace: SimpleEssentials.DataStore
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#

```csharp
public bool Add<T>(
    T obj
)
```

VB

```vbnet
Public Function Add(Of T) ( _
    obj As T _
) As Boolean
```

C++

```cpp
public:
    template<typename T>
    bool^ Add(  
        T obj
    )
```

Type Parameters

T

Parameters

obj
    Type: T
See Also

DbStore Class
SimpleEssentials.DataStore Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
DbStore..::..AddAndReturnId<Of <('T')>> Method

Namespace: SimpleEssentials.DataStore
Assembly: SimpleEssentials (in SimpleEssentials.dll)
## Syntax

**C#**

```csharp
public int AddAndReturnId<T>(
    string sql,
    T obj
)
```

**VB**

```vbnet
Public Function AddAndReturnId(Of T) ( _
    sql As String, _
    obj As T _
) As Integer
```

**C++**

```cpp
public:

generic<
typename T>
int^ AddAndReturnId(
    String^ sql,
    T obj
)
```

### Type Parameters

**T**

### Parameters

**sql**
- Type: String

**obj**
- Type: T
See Also

DbStore Class
SimpleEssentials.DataStore Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
DbStore..::..AddList<(Of (('T')>>))>

Method

Namespace: SimpleEssentials.DataStore
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

**C#**

```csharp
public int AddList<T>(
    IEnumerable<T> obj,
    string sql
)
```

**VB**

```vbnet
Public Function AddList(Of T) (_
    obj As IEnumerable(Of T), _
    sql As String _
) As Integer
```

**C++**

```cpp
public:
    template<typename T>
    int AddList(
        IEnumerable<T> obj,
        String sql
    )
```

**Type Parameters**

T

**Parameters**

**obj**

Type: `IEnumerable<((Of (<<T>>)>)`

**sql**

Type: `String`
See Also

DbStore Class
SimpleEssentials.DataStore Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
DbStore::<: BulkInsert<(Of <('T')>)> Method

Namespace:  SimpleEssentials.DataStore
Assembly:  SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#

```csharp
public void BulkInsert<T>(
    IEnumerable<T> obj,
    string tableName
)
```

VB

```vbnet
Public Sub BulkInsert(Of T) ( _
    obj As IEnumerable(Of T), _
    tableName As String _
)
```

C++

```c++
public:
    template<typename T>
    void BulkInsert(
        IEnumerable<T>^ obj,
        String^ tableName
    )
```

Type Parameters

T

Parameters

obj
    Type: IEnumerable<(Of (Of T)>)

tableName
    Type: String
See Also

DbStore Class
SimpleEssentials.DataStore Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
DbStore...::..Delete<(Of <("T")>>)>
Method

Namespace:  SimpleEssentials.DataStore
Assembly:  SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#

public bool Delete<T>(
    T obj
)

VB

Public Function Delete(Of T) ( _
    obj As T _
) As Boolean

C++

public:
    template<typename T>
    bool^ Delete(  
        T obj       
    )

Type Parameters

T

Parameters

obj
    Type: T
See Also

DbStore Class
SimpleEssentials.DataStore Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
DbStore...Execute Method

Namespace:  SimpleEssentials.DataStore
Assembly:  SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#

```csharp
public int Execute(
        string sql,
        Object param
    )
```

VB

```vbnet
Public Function Execute (
    sql As String,
    param As Object
) As Integer
```

C++

```cpp
public: int Execute(
        String^ sql,
        Object^ param
    )
```

Parameters

sql
  Type: String

param
  Type: Object
See Also

DbStore Class
SimpleEssentials.DataStore Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
DbStore..::..ExecuteScalar Method

Namespace:  SimpleEssentials.DataStore
Assembly:  SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#

```csharp
public int ExecuteScalar(
    string sql,
    Object param
)
```

VB

```vbnet
Public Function ExecuteScalar ( _
    sql As String, _
    param As Object _
) As Integer
```

C++

```cpp
public: int^ ExecuteScalar(
    String^ sql,
    Object^ param
)
```

Parameters

sql
Type: String

param
Type: Object
See Also

- DbStore Class
- SimpleEssentials.DataStore Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
DbStore..::..Get(Of<'T'>)>

**Method**

**Namespace:**  [SimpleEssentials.DataStore](#)

**Assembly:**  SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#

VB

C++

public T Get<T>(
    Object id
)

Public Function Get(Of T) ( _
    id As Object _
) As T

public:
    generic<typename T>
    T Get(
        Object^ id
    )

Type Parameters

T

Parameters

id
    Type: Object
See Also

DbStore Class
SimpleEssentials.DataStore Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
DbStore...::.GetByParameters<(Of<br>"T")>) Method

Namespace: SimpleEssentials, DataStore
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#

```csharp
public IEnumerable<T> GetByParameters<T>(
    string sql,
    Object param
)
```

VB

```vbnet
Public Function GetByParameters(Of T) ( _
    sql As String, _
    param As Object _
) As IEnumerable(Of T)
```

C++

```csharp
public: generic<typename T>
    IEnumerable<T>^ GetByParameters(
        String^ sql,
        Object^ param
    )
```

Type Parameters

T

Parameters

sql
  Type: String

param
  Type: Object
See Also

DbStore Class
SimpleEssentials.DataStore Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
DbStore...:::GetByType<(Of <('<T>')>)> Method

Namespace: SimpleEssentials.DataStore
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#

public IEnumerable<T> GetByType<T>()

VB

Public Function GetByType(Of T) As IEnumerable(Of T)

c++

public

generic<typename T>

IEnumerable<T>^ GetByType()

Type Parameters

T
See Also

DbStore Class
SimpleEssentials.DataStore Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
DbStore...::..GetMultiMap<(Of <('T, T2>>)>) Method (String, Func<(Of <('T, T2, T)>>)>, Object, String)

Namespace: SimpleEssentials.DataStore
Assembly: SimpleEssentials (in SimpleEssentials.dll)
**Syntax**

**C#**

```csharp
public IEnumerable<T> GetMultiMap<T, T2>(
    string sql,
    Func<T, T2, T> func,
    Object param,
    string splitOn
)
```

**VB**

```vbnet
Public Function GetMultiMap(Of T, T2) ( _
    sql As String, _
    func As Func(Of T, T2, T), _
    param As Object, _
    splitOn As String _
) As IEnumerable(Of T)
```

**C++**

```cpp
public:
generic<typename T, typename T2>
IEquatable<T>^ GetMultiMap(
    String^ sql,
    Func<T, T2, T>^ func,
    Object^ param,
    String^ splitOn
)
```

**Type Parameters**

- **T**
- **T2**

**Parameters**

- **sql**
  - Type: **String**

- **func**
Type: Func<(Of <('T, T2, T)>)>  

param  
  Type: Object  

splitOn  
  Type: String
See Also

DbStore Class
GetMultiMap Overload
SimpleEssentials.DataStore Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
DbStore..::.GetMultiMap<(Of <("T, T2, T3">)>)> Method (String, Func<(Of <("T, T2, T3, T">)>)>, Object, String)

Namespace: SimpleEssentials.DataStore
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#

```csharp
public IEnumerable<T> GetMultiMap<T, T2, T3>(
    string sql,
    Func<T, T2, T3, T> func,
    Object param,
    string splitOn
)
```

VB

```vbnet
Public Function GetMultiMap(Of T, T2, T3) ( _
    sql As String, _
    func As Func(Of T, T2, T3, T), _
    param As Object, _
    splitOn As String _
) As IEnumerable(Of T)
```

C++

```cpp
public:

generic<typename T, typename T2, typename T3>
IEnumerable<T>^ GetMultiMap(
    String^ sql,
    Func<T, T2, T3, T>^ func,
    Object^ param,
    String^ splitOn
)
```

Type Parameters

T
T2
T3

Parameters

sql

Type: String
func
    Type: Func<
        Of
        <
            T
            ,
            T2
            ,
            T3
            ,
            T
        >
    >

param
    Type: Object

splitOn
    Type: String
See Also

DbStore Class
GetMultiMap Overload
SimpleEssentials.DataStore Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
DbStore..::..GetMultiMap<Of <('T, T2, T3, T4)>>) Method (String, Func<Of <('T, T2, T3, T4, T)>>, Object, String)

Namespace: SimpleEssentials.DataStore
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#

public IEnumerable<T> GetMultiMap<T, T2, T3, T4>(
    string sql,
    Func<T, T2, T3, T4, T> func,
    Object param,
    string splitOn
)

Public Function GetMultiMap(Of T, T2, T3, T4) ( _
    sql As String, _
    func As Func(Of T, T2, T3, T4, T), _
    param As Object, _
    splitOn As String _
) As IEnumerable(Of T)

public:
    generic<typename T, typename T2, typename T3, typename T4>
    IEnumerable<T>^ GetMultiMap(
        String^ sql,
        Func<T, T2, T3, T4, T>^ func,
        Object^ param,
        String^ splitOn
    )

Type Parameters

T
T2
T3
T4

Parameters

sql
    Type: String
func
    Type: Func<(<`T, T2, T3, T4, T`)>)

param
    Type: Object

splitOn
    Type: String
See Also

DbStore Class
GetMultiMap Overload
SimpleEssentials.DataStore Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
DbStore...:::GetMultiMap<(Of <(%'T, T2, T3, T4, T5>>)>) Method (String, Func<(Of <(%'T, T2, T3, T4, T5, T>>)>>, Object, String)

Namespace: SimpleEssentials.DataStore
Assembly: SimpleEssentials (in SimpleEssentials.dll)
## Syntax

### C#

```csharp
public IEnumerable<T> GetMultiMap<T, T2, T3, T4, T5>(
    string sql,
    Func<T, T2, T3, T4, T5, T> func,
    Object param,
    string splitOn
)
```

### VB

```vbnet
Public Function GetMultiMap(Of T, T2, T3, T4, T5) ( _
    sql As String, _
    func As Func(Of T, T2, T3, T4, T5, T), _
    param As Object, _
    splitOn As String _
) As IEnumerable(Of T)
```

### C++

```cpp
public:
    template<typename T, typename T2, typename T3, typename T4, typename T5>
    IEnumerable<T>^ GetMultiMap(
        String^ sql,
        Func<T, T2, T3, T4, T5, T>^ func,
        Object^ param,
        String^ splitOn
    )
```

### Type Parameters

- **T**
- **T2**
- **T3**
- **T4**
- **T5**

### Parameters

- **sql**
Type: **String**

```latex
func
   Type: Func<(<T, T2, T3, T4, T5, T>)>

param
   Type: **Object**

splitOn
   Type: **String**
See Also

DbStore Class
GetMultiMap Overload
SimpleEssentials.DataStore Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
DbStore...:::GetMultiMap<Of <('T, T2, T3, T4, T5, T6)>>) Method (String, Func<Of <('T, T2, T3, T4, T5, T6, T)>>>, Object, String)

Namespace: SimpleEssentials.DataStore
Assembly: SimpleEssentials (in SimpleEssentials.dll)
**Syntax**

**C#**
```csharp
public IEnumerable<T> GetMultiMap<T, T2, T3, T4, T5, T6>(
    string sql,
    Func<T, T2, T3, T4, T5, T6, T> func,
    Object param,
    string splitOn
)
```

**VB**
```vbnet
Public Function GetMultiMap(Of T, T2, T3, T4, T5, T6)(
    _
    sql As String,
    _
    func As Func(Of T, T2, T3, T4, T5, T6, T),
    _
    param As Object,
    _
    splitOn As String
) As IEnumerable(Of T)
```

**C++**
```cpp
public:
    template<typename T, typename T2, typename T3, typename T4, typename T5, typename T6>
    IEnumerable<T>^ GetMultiMap(
        String^ sql,
        Func<T, T2, T3, T4, T5, T6, T>^ func,
        Object^ param,
        String^ splitOn
    )
```

**Type Parameters**

- T
- T2
- T3
- T4
- T5
- T6

**Parameters**
sql
  Type: String

func
  Type: Func<(Of (<'T, T2, T3, T4, T5, T6, T>))>

param
  Type: Object

splitOn
  Type: String
See Also

DbStore Class
GetMultiMap Overload
SimpleEssentials.DataStore Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
DbStore..:::Update(Of (<'T'>))>

Method

Namespace:  SimpleEssentials.DataStore
Assembly:  SimpleEssentials (in SimpleEssentials.dll)
## Syntax

### C#

```csharp
public bool Update<T>(
    T obj
)
```

### VB

```vbnet
Public Function Update(Of T) ( _
    obj As T _
) As Boolean
```

### C++

```cpp
public:
    template<typename T>
    bool Update(
        T obj
    )
```

### Type Parameters

- **T**

### Parameters

- **obj**
  - Type: **T**
See Also

DbStore Class
SimpleEssentials.DataStore Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
IDataStore Interface

Namespace: SimpleEssentials.DataStore
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#
VB
C++

public interface IDataStore
Public Interface IDataStore
public interface class IDataStore
See Also

IDataStore Members
SimpleEssentials.DataStore Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
IDataStore Members

The IDataStore type exposes the following members.
### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add&lt;Of &lt;&lt;(T)&gt;&gt;</td>
<td></td>
</tr>
<tr>
<td>AddAndReturnId&lt;Of &lt;&lt;(T)&gt;&gt;</td>
<td></td>
</tr>
<tr>
<td>AddList&lt;Of &lt;&lt;(T)&gt;&gt;</td>
<td></td>
</tr>
<tr>
<td>BulkInsert&lt;Of &lt;&lt;(T)&gt;&gt;</td>
<td></td>
</tr>
<tr>
<td>Delete&lt;Of &lt;&lt;(T)&gt;&gt;</td>
<td></td>
</tr>
<tr>
<td>Equals(System.Object)</td>
<td>Determines whether the specified object is equal to the current object.</td>
</tr>
<tr>
<td>Execute</td>
<td>(Inherited from <strong>Object</strong>.)</td>
</tr>
<tr>
<td>ExecuteScalar</td>
<td></td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection.</td>
</tr>
<tr>
<td>Get&lt;Of &lt;&lt;(T)&gt;&gt;</td>
<td>(Inherited from <strong>Object</strong>.)</td>
</tr>
<tr>
<td>GetByParameters&lt;Of &lt;&lt;(T)&gt;&gt;</td>
<td></td>
</tr>
<tr>
<td>GetByType&lt;Of &lt;&lt;(T)&gt;&gt;</td>
<td></td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from <strong>Object</strong>.)</td>
</tr>
<tr>
<td>GetMultiMap&lt;Of &lt;&lt;(T, T2)&gt;&gt;</td>
<td></td>
</tr>
<tr>
<td>(String, Func&lt;Of &lt;&lt;(T, T2, T)&gt;&gt;)</td>
<td></td>
</tr>
<tr>
<td>Object, String</td>
<td></td>
</tr>
<tr>
<td>GetMultiMap&lt;Of &lt;&lt;(T, T2, T3)&gt;&gt;</td>
<td></td>
</tr>
<tr>
<td>(String, Func&lt;Of &lt;&lt;(T, T2, T3, T)&gt;&gt;)</td>
<td></td>
</tr>
<tr>
<td>Object, String</td>
<td></td>
</tr>
<tr>
<td>GetMultiMap&lt;Of &lt;&lt;(T, T2, T3, T4)&gt;&gt;</td>
<td></td>
</tr>
<tr>
<td>(String, Func&lt;Of &lt;&lt;(T, T2, T3, T4, T)&gt;&gt;)</td>
<td></td>
</tr>
<tr>
<td>Object, String</td>
<td></td>
</tr>
<tr>
<td>GetMultiMap&lt;Of &lt;&lt;(T, T2, T3, T4, T)&gt;&gt;</td>
<td></td>
</tr>
</tbody>
</table>
T5>>>(String, Func<Of <<'(T, T2, T3, T4, T5, T)>>>, Object, String)

GetMultiMap<Of <<'(T, T2, T3, T4, T5, T6)>>>(String, Func<Of <<'(T, T2, T3, T4, T5, T6, T)>>>, Object, String)

GetType

MemberwiseClone

ToString

Update<Of <<'(T)>>>

Gets the Type of the current instance. (Inherited from Object.)

Creates a shallow copy of the current Object. (Inherited from Object.)

Returns a string that represents the current object. (Inherited from Object.)
See Also

IDataStore Interface
SimpleEssentials.DataStore Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
IDataStore Methods

The IDataStore type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add&lt;(&lt;T&gt;&gt;)&gt;</td>
<td>Determines whether the specified object is equal to the current object.</td>
</tr>
<tr>
<td>AddAndReturnId&lt;(&lt;T&gt;&gt;)&gt;</td>
<td>(Inherited from <a href="#">Object</a>)</td>
</tr>
<tr>
<td>AddList&lt;(&lt;T&gt;&gt;)&gt;</td>
<td></td>
</tr>
<tr>
<td>BulkInsert&lt;(&lt;T&gt;&gt;)&gt;</td>
<td></td>
</tr>
<tr>
<td>Delete&lt;(&lt;T&gt;&gt;)&gt;</td>
<td></td>
</tr>
<tr>
<td>Equals(System.Object)</td>
<td></td>
</tr>
<tr>
<td>Execute</td>
<td>Allows an object to try to free resources and perform other cleanup</td>
</tr>
<tr>
<td>ExecuteScalar</td>
<td>operations before it is reclaimed by garbage collection.</td>
</tr>
<tr>
<td>Finalize</td>
<td>(Inherited from <a href="#">Object</a>)</td>
</tr>
<tr>
<td>Get&lt;(&lt;T&gt;&gt;)&gt;</td>
<td>Serves as the default hash function.</td>
</tr>
<tr>
<td>GetByParameters&lt;(&lt;T&gt;&gt;)&gt;</td>
<td>(Inherited from <a href="#">Object</a>)</td>
</tr>
<tr>
<td>GetByType&lt;(&lt;T&gt;&gt;)&gt;</td>
<td></td>
</tr>
<tr>
<td>GetHashCode</td>
<td></td>
</tr>
<tr>
<td>GetMultiMap&lt;(&lt;T, T2&gt;&gt;)&gt;</td>
<td></td>
</tr>
<tr>
<td>(String, Func&lt;(&lt;T, T2, T&gt;&gt;), Object, String)</td>
<td></td>
</tr>
<tr>
<td>GetMultiMap&lt;(&lt;T, T2&gt;&gt;)(String, Func&lt;(&lt;T, T2, T3, T4&gt;&gt;), Object, String)</td>
<td></td>
</tr>
<tr>
<td>GetMultiMap&lt;(&lt;T, T2, T3&gt;&gt;)(String, Func&lt;(&lt;T, T2, T3, T4, T&gt;&gt;), Object, String)</td>
<td></td>
</tr>
<tr>
<td>GetMultiMap&lt;(&lt;T, T2, T3, T4&gt;&gt;)&gt;</td>
<td></td>
</tr>
</tbody>
</table>
GetMultiMap<(Of <<'(T, T2, T3, T4, T5, T6)>), String, Func<(Of <<'(T, T2, T3, T4, T5, T6, T)>), Object, String)>

GetMultiMap<(Of <<'(T, T2, T3, T4, T5, T6)>), String, Func<(Of <<'(T, T2, T3, T4, T5, T6, T)>), Object, String)>

GetType

MemberwiseClone

ToString

Update<(Of <<'(T)>)>>

Gets the **Type** of the current instance. (Inherited from **Object**.)

Creates a shallow copy of the current **Object**. (Inherited from **Object**.)

Returns a string that represents the current object. (Inherited from **Object**.)
See Also

IDataStore Interface
SimpleEssentials.DataStore Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
IDataStore..::.Add(Of ("T")<>)

Method

Namespace: SimpleEssentials.DataStore
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#

VB

C++

```csharp
bool Add<T>(
    T obj
)
```

```vbnet
Function Add(Of T) ( _
    obj As T _
) As Boolean
```

generic<typename T>

```cpp
bool^ Add(
    T obj
)
```

**Type Parameters**

T

**Parameters**

obj

    Type: T
See Also

IDataStore Interface
SimpleEssentials.DataStore Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
IDataStore::..AddAndReturnId<Of <('T')>> Method

Namespace: SimpleEssentials.DataStore
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#

```csharp
int AddAndReturnId<T>(
    string sql,
    T obj
)
```

VB

```vbnet
Function AddAndReturnId(Of T) ( _
    sql As String, _
    obj As T _
) As Integer

generic<typename T>

```vbnet
int^ AddAndReturnId(
    String^ sql,
    T obj
)
```

C++

```cpp
int AddAndReturnId<T>(
    string sql,
    T obj
)
```

Type Parameters

T

Parameters

sql
  Type: String

obj
  Type: T
See Also

IDataStore Interface
SimpleEssentials.DataStore Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
IDataStore..::.AddList<(Of <('T)>)> Method

Namespace: SimpleEssentials.DataStore
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#
VB
C++

```c
int AddList<T>(
    IEnumerable<T> obj,
    string sql
)
```

Function AddList(Of T) (_
    obj As IEnumerable(Of T), _
    sql As String _
) As Integer

generic<typename T>

```c
int^ AddList(
    IEnumerable^ obj,
    String^ sql
)
```

Type Parameters

T

Parameters

obj
    Type: IEnumerable<(Of (Of<T>)>)

sql
    Type: String
See Also

IDataStore Interface
SimpleEssentials.DataStore Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
IDataStore..::..BulkInsert<(Of <('T')>)> Method

Namespace: SimpleEssentials.DataStore
Assembly: SimpleEssentials (in SimpleEssentials.dll)
### Syntax

**C#**
```csharp
void BulkInsert<T>(
    IEnumerable<T> obj,
    string tableName
)
```

**VB**
```vbnet
Sub BulkInsert(Of T) ( _
    obj As IEnumerable(Of T), _
    tableName As String _
)
```

**C++**
```cpp
generic<typename T>
void BulkInsert(
    IEnumerable<T>^ obj,
    String^ tableName
)
```

### Type Parameters

**T**

### Parameters

**obj**
- **Type:** `IEnumerable<Of ((T)>)`

**tableName**
- **Type:** `String`
See Also

IDataStore Interface
SimpleEssentials.DataStore Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
IDataStore..::..Delete(Of (<'T'>))>
Method

Namespace: SimpleEssentials.DataStore
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#

bool Delete<T>(
    T obj
)

Function Delete(Of T) ( _
    obj As T _
) As Boolean

generic<typename T>
bool^ Delete(
    T obj
)

Type Parameters

T

Parameters

obj
    Type: T
See Also

IDataStore Interface
SimpleEssentials.DataStore Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
IDataStore..::..Execute Method

Namespace: SimpleEssentials.DataStore
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#
VB
C++

```csharp
int Execute(
    string sql,
    Object param
)
```

```vbnet
Function Execute (_
    sql As String, _
    param As Object _
) As Integer
```

```cpp
int^ Execute( 
    String^ sql, 
    Object^ param
)
```

**Parameters**

sql
- Type: String

param
- Type: Object
See Also

IDataStore Interface
SimpleEssentials.DataStore Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
IDataStore..::.ExecuteScalar Method

Namespace: SimpleEssentials.DataStore
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#

```csharp
int ExecuteScalar(
    string sql,
    Object param
)
```

VB

```vbnet
Function ExecuteScalar ( _
    sql As String, _
    param As Object _) _
) As Integer
```

C++

```cpp
int^ ExecuteScalar(
    String^ sql,
    Object^ param
)
```

Parameters

- sql
  Type: String

- param
  Type: Object
See Also

IDataStore Interface
SimpleEssentials.DataStore Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
IDataStore..::.Get(Of '<T'>)}

Method

Namespace: SimpleEssentials.DataStore
Assembly: SimpleEssentials (in SimpleEssentials.dll)
**Syntax**

C#  
VB  
C++

```
T Get<T>(
    Object id
)

Function Get(Of T) ( _
    id As Object _
) As T

generic<typename T>
T Get(
    Object^ id
)
```

**Type Parameters**

T

**Parameters**

id  

Type: Object
See Also

IDataStore Interface
SimpleEssentials.DataStore Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
IDataStore..::.GetByParameters(Of <'T'>) Method

Namespace: SimpleEssentials.DataStore
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#
VB
C++

IEnumerator<T> GetByParameters<T>(
        string sql,
        Object param
    )

Function GetByParameters(Of T) ( _
    sql As String, _
    param As Object _
) As IEnumerable(Of T)

generic<typename T>
IEnumerator<T>^ GetByParameters(
    String^ sql,
    Object^ param
    )

Type Parameters

T

Parameters

sql
    Type: String

param
    Type: Object
See Also

IDataStore Interface
SimpleEssentials.DataStore Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
IDataStore..::.GetByType<Of<br> ‎((<'T>))> Method<br><br>Namespace: SimpleEssentials.DataStore<br>Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#

```csharp
IEnumerable<T> GetByType<T>()
```

VB

```vbnet
Function GetByType(Of T) As IEnumerable(Of T)
```

c++

```csharp
generic<typename T>
IEnumerable<T>^ GetByType()
```

**Type Parameters**

T
See Also

IDataStore Interface
SimpleEssentials.DataStore Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
IDataStore:::..GetMultiMap<Of <('T, T2)> > Method (String, Func<Of <('T, T2, T)> >, Object, String)

Namespace: SimpleEssentials.DataStore
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

**C#**

```csharp
IEnumerable<T> GetMultiMap<T, T2>(
    string sql,
    Func<T, T2, T> func,
    Object param,
    string splitOn
)
```

**VB**

```vbnet
Function GetMultiMap(Of T, T2) ( _
    sql As String, _
    func As Func(Of T, T2, T), _
    param As Object, _
    splitOn As String _
) As IEnumerable(Of T)
```

generic<typename T, typename T2>

```csharp
IEnumerable<T> GetMultiMap(
    string^ sql,
    Func<T, T2, T>^ func,
    Object^ param,
    String^ splitOn
)
```

**Type Parameters**

- T
- T2

**Parameters**

- **sql**
  - Type: **String**

- **func**
  - Type: Func<(Of ((<`T`, T2>, T)>)>
param
  Type: Object

splitOn
  Type: String
See Also

IDataStore Interface
GetMultiMap Overload
SimpleEssentials.DataStore Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
IDataStore..::..GetMultiMap<(Of <(<'T, T2, T3'>)>))> Method (String, Func<(Of <(<'T, T2, T3, T'>)>)), Object, String)

Namespace: SimpleEssentials.DataStore
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#
VB
C++

```csharp
IEnumerable<T> GetMultiMap<T, T2, T3>(
    string sql,
    Func<T, T2, T3, T> func,
    Object param,
    string splitOn
)
```

Function GetMultiMap(Of T, T2, T3) ( _
    sql As String, _
    func As Func(Of T, T2, T3, T), _
    param As Object, _
    splitOn As String _
) As IEnumerable(Of T)

generic<typename T, typename T2, typename T3>
```csharp
IEnumerable<T>^ GetMultiMap(
    String^ sql,
    Func<T, T2, T3, T>^ func,
    Object^ param,
    String^ splitOn
)
```

Type Parameters

T
T2
T3

Parameters

sql
    Type: String

func
Type: Func<(Of <('T, T2, T3, T)>))>

param
Type: Object

splitOn
Type: String
See Also

IDataStore Interface
GetMultiMap Overload
SimpleEssentials.DataStore Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
IDataStore::GetMultiMap<Of <('T, T2, T3, T4)> > Method (String, Func<Of <('T, T2, T3, T4, T)> >, Object, String)

Namespace: SimpleEssentials.DataStore
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#

```csharp
IEnumerable<T> GetMultiMap<T, T2, T3, T4>(
    string sql,
    Func<T, T2, T3, T4, T> func,
    Object param,
    string splitOn
)
```

VB

```vbnet
Function GetMultiMap(Of T, T2, T3, T4) ( _
    sql As String, _
    func As Func(Of T, T2, T3, T4, T), _
    param As Object, _
    splitOn As String _
) As IEnumerable(Of T)
```

C++

```cpp
generic<typename T, typename T2, typename T3, typename T4>
IEnumerable<T>^ GetMultiMap(
    String^ sql,
    Func<T, T2, T3, T4, T>^ func,
    Object^ param,
    String^ splitOn
)
```

Type Parameters

T
T2
T3
T4

Parameters

sql

Type: String
func
   Type: Func<(Of <(T, T2, T3, T4, T)>)>  

param
   Type: Object

splitOn
   Type: String
See Also

IDataStore Interface
GetMultiMap Overload
SimpleEssentials.DataStore Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
IDataStore...::..GetMultiMap<($(typeof T, T2, T3, T4, T5))> Method (String, Func<($(typeof T, T2, T3, T4, T5, T))>, Object, String)

Namespace: SimpleEssentials.DataStore
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#

VB

C++

```csharp
IEnumerable<T> GetMultiMap<T, T2, T3, T4, T5>(
    string sql,
    Func<T, T2, T3, T4, T5, T> func,
    Object param,
    string splitOn
)
```

```vb
Function GetMultiMap(Of T, T2, T3, T4, T5) (_
    sql As String, _
    func As Func(Of T, T2, T3, T4, T5, T), _
    param As Object, _
    splitOn As String _
) As IEnumerable(Of T)
```

```cpp
generic<typename T, typename T2, typename T3, typename T4, typename T5> GetMultiMap(
    String^ sql,
    Func<T, T2, T3, T4, T5, T>^ func,
    Object^ param,
    String^ splitOn
)
```

Type Parameters

T
T2
T3
T4
T5

Parameters

sql

Type: String
func
   Type: Func<((T, T2, T3, T4, T5, T)>)>

param
   Type: Object

splitOn
   Type: String
See Also

IDataStore Interface
GetMultiMap Overload
SimpleEssentials.DataStore Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
IDataStore..::.GetMultiMap<(Of (<'T, T2, T3, T4, T5, T6>))> Method (String, Func<(Of (<'T, T2, T3, T4, T5, T6, T>))>, Object, String)

Namespace: SimpleEssentials.DataStore
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#

VB

C++

```csharp
IEnumerable<T> GetMultiMap<T, T2, T3, T4, T5, T6>(
    string sql,
    Func<T, T2, T3, T4, T5, T6, T> func,
    Object param,
    string splitOn
)
```

Function GetMultiMap(Of T, T2, T3, T4, T5, T6) ( _
    sql As String, _
    func As Func(Of T, T2, T3, T4, T5, T6, T), _
    param As Object, _
    splitOn As String _
) As IEnumerable(Of T)

generic<typename T, typename T2, typename T3, typename T4, typename T5, typename T6>
IEnumerable<T>& GetMultiMap(
    String& sql,
    Func<T, T2, T3, T4, T5, T6, T>& func,
    Object& param,
    String& splitOn
)
```

Type Parameters

T
T2
T3
T4
T5
T6

Parameters

sql
Type: String

func
  Type: Func<(<T, T2, T3, T4, T5, T6, T>)>

param
  Type: Object

splitOn
  Type: String
See Also

IDataStore Interface
GetMultiMap Overload
SimpleEssentials.DataStore Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
IDataStore..::..Update<(Of <(<'T'>)>))> Method

Namespace: SimpleEssentials.DataStore
Assembly: SimpleEssentials (in SimpleEssentials.dll)
**Syntax**

---

**C#**

```csharp
bool Update<T>(
    T obj
)
```

**VB**

```vbnet
Function Update(Of T) ( _
    obj As T _
) As Boolean
```

generic<typename T>

```csharp
bool^ Update(
    T obj
)
```

**Type Parameters**

T

**Parameters**

obj

    Type: T
See Also

IDataStore Interface
SimpleEssentials.DataStore Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
IDbStore Interface

**Namespace:**  [SimpleEssentials.DataStore](#)

**Assembly:**  SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#
VB
C++

public interface IDbStore : IDataStore

Public Interface IDbStore _
  Inherits IDataStore

public interface class IDbStore : IDataStore
See Also

IDbStore Members
SimpleEssentials.DataStore Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
IDbStore Members

The IDbStore type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add&lt;Of &lt;&lt;'T'&gt;&gt;&gt;()</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>AddAndReturnId&lt;Of &lt;&lt;'T'&gt;&gt;&gt;()</td>
<td></td>
</tr>
<tr>
<td>AddList&lt;Of &lt;&lt;'T'&gt;&gt;&gt;()</td>
<td></td>
</tr>
<tr>
<td>BulkInsert&lt;Of &lt;&lt;'T'&gt;&gt;&gt;()</td>
<td></td>
</tr>
<tr>
<td>Delete&lt;Of &lt;&lt;'T'&gt;&gt;&gt;()</td>
<td></td>
</tr>
<tr>
<td>Equals(System.Object)</td>
<td></td>
</tr>
<tr>
<td>Execute</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>ExecuteScalar</td>
<td></td>
</tr>
<tr>
<td>Finalize</td>
<td></td>
</tr>
<tr>
<td>Get&lt;Of &lt;&lt;'T'&gt;&gt;&gt;()</td>
<td></td>
</tr>
<tr>
<td>GetByIdParameters&lt;Of &lt;&lt;'T'&gt;&gt;&gt;()</td>
<td></td>
</tr>
<tr>
<td>GetByIdType&lt;Of &lt;&lt;'T'&gt;&gt;&gt;()</td>
<td></td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>GetMultiMap&lt;Of &lt;&lt;'T', T2', T3'&gt;&gt;&gt;()&lt;br&gt;(String, Func&lt;Of &lt;&lt;'T', T2, T3'&gt;&gt;&gt;()&lt;br&gt;Object, String)</td>
<td></td>
</tr>
<tr>
<td>GetMultiMap&lt;Of &lt;&lt;'T', T2, T3'&gt;&gt;()&lt;br&gt;(String, Func&lt;Of &lt;&lt;'T', T2, T3, T4'&gt;&gt;&gt;()&lt;br&gt;Object, String)</td>
<td></td>
</tr>
<tr>
<td>GetMultiMap&lt;Of &lt;&lt;'T', T2, T3, T4'&gt;&gt;()&lt;br&gt;(String, Func&lt;Of &lt;&lt;'T', T2, T3, T4, T'&gt;&gt;&gt;()&lt;br&gt;Object, String)</td>
<td></td>
</tr>
</tbody>
</table>
GetMultiMap<(Of <<'(T, T2, T3, T4, T5, T6)>>)(String, Func<(Of <<'(T, T2, T3, T4, T5, T6)>>, Object, String)

Gets the **Type** of the current instance.  
(Inherited from **Object**.)

`MemberwiseClone` 
Creates a shallow copy of the current **Object**.  
(Inherited from **Object**.)

`ToString` 
Returns a string that represents the current object.  
(Inherited from **Object**.)

`Update<(Of <<'(T)>>)`
See Also

IDbStore Interface
SimpleEssentials.DataStore Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
IDbStore Methods

The IDbStore type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add&lt;(Of &lt;&lt;(T)&gt;&gt;)</td>
<td>Adds an item to the collection.</td>
</tr>
<tr>
<td>AddAndReturnId&lt;(Of &lt;&lt;(T)&gt;&gt;)</td>
<td>Adds an item to the collection and returns its ID.</td>
</tr>
<tr>
<td>AddList&lt;(Of &lt;&lt;(T)&gt;&gt;)</td>
<td>Adds a list of items to the collection.</td>
</tr>
<tr>
<td>BulkInsert&lt;(Of &lt;&lt;(T)&gt;&gt;)</td>
<td>Inserts multiple items into the collection.</td>
</tr>
<tr>
<td>Delete&lt;(Of &lt;&lt;(T)&gt;&gt;)</td>
<td>Removes an item from the collection.</td>
</tr>
<tr>
<td>Equals(System.Object)</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>Execute</td>
<td>Performs an operation on the collection.</td>
</tr>
<tr>
<td>ExecuteScalar</td>
<td>Returns a single value from the collection.</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>Get&lt;(Of &lt;&lt;(T)&gt;&gt;)</td>
<td>Gets an item from the collection.</td>
</tr>
<tr>
<td>GetByParameters&lt;(Of &lt;&lt;(T)&gt;&gt;)</td>
<td>Gets an item from the collection using parameters.</td>
</tr>
<tr>
<td>GetByType&lt;(Of &lt;&lt;(T)&gt;&gt;)</td>
<td>Gets an item from the collection by type.</td>
</tr>
<tr>
<td>GetMultiMap&lt;(Of &lt;&lt;(T, T2)&gt;&gt;)</td>
<td>Gets a multi-map from the collection.</td>
</tr>
<tr>
<td>(String, Func&lt;(Of &lt;&lt;(T, T2, T)&gt;&gt;&gt;, Object, String))</td>
<td></td>
</tr>
<tr>
<td>GetMultiMap&lt;(Of &lt;&lt;(T, T2, T3)&gt;&gt;)</td>
<td>Gets a multi-map from the collection.</td>
</tr>
<tr>
<td>(String, Func&lt;(Of &lt;&lt;(T, T2, T3, T4)&gt;&gt;, Object, String))</td>
<td></td>
</tr>
<tr>
<td>GetMultiMap&lt;(Of &lt;&lt;(T, T2, T3, T4)&gt;&gt;)</td>
<td>Gets a multi-map from the collection.</td>
</tr>
<tr>
<td>(String, Func&lt;(Of &lt;&lt;(T, T2, T3, T4, T)&gt;&gt;, Object, String))</td>
<td></td>
</tr>
</tbody>
</table>
T5>>>(String, Func<Of <<-('T, T2, T3, T4, T5, T>>>), Object, String)
GetMultiMap<Of <<-('T, T2, T3, T4, T5, T6>>>)(String, Func<Of <<-('T, T2, T3, T4, T5, T6, T>>>), Object, String)

GetType

MemberwiseClone

ToString

Update<Of <<-('T)>>>

Gets the Type of the current instance.
(Inherited from Object.)
Creates a shallow copy of the current Object.
(Inherited from Object.)
Returns a string that represents the current object.
(Inherited from Object.)
See Also

IDbStore Interface
SimpleEssentials.DataStore Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
IDbStore..::..Add<(Of <('T')?>)>)

Method

Namespace: SimpleEssentials.DataStore
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#
VB
C++

bool Add<T>(
    T obj
)

Function Add(Of T)( _
    obj As T _
) As Boolean

generic<typename T>
bool^ Add(
    T obj
)

Type Parameters

T

Parameters

obj
    Type: T
See Also

IDbStore Interface
SimpleEssentials.DataStore Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
IDbStore..::..AddAndReturnId<(Of <('T')>)> Method

Namespace:  SimpleEssentials.DataStore
Assembly:  SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#

```csharp
int AddAndReturnId<T>(
    string sql,
    T obj
)
```

VB

```vbnet
Function AddAndReturnId(Of T) ( _
    sql As String, _
    obj As T _
) As Integer

generic<typename T>
int^ AddAndReturnId(
    String^ sql,
    T obj
)
```

C++

```cpp
int AddAndReturnId<T>(
    string sql,
    T obj
)
```

Type Parameters

T

Parameters

sql
Type: String

obj
Type: T
See Also

IDbStore Interface
SimpleEssentials.DataStore Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
IDbStore..::..AddList(Of ("T")>)
Method

Namespace: SimpleEssentials.DataStore
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#
VB
C++

```csharp
int AddList<T>(
    IEnumerable<T> obj,
    string sql
)
```

Function AddList(Of T) ( _
    obj As IEnumerable(Of T), _
    sql As String _
) As Integer

generic<typename T>

```csharp
int^ AddList(
    IEnumerable<T>^ obj,
    String^ sql
)
```

Type Parameters

T

Parameters

obj
    Type: IEnumerable<(Of <(T)>)>^

sql
    Type: String
See Also

IDbStore Interface
SimpleEssentials.DataStore Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
IDbStore..::..BulkInsert<(Of <('T')>)> Method

Namespace: SimpleEssentials.DataStore
Assembly: SimpleEssentials (in SimpleEssentials.dll)
**Syntax**

C#  
```
voidBulkInsert<T>(
    IEnumerable<T> obj,
    string tableName
)
```

VB  
```
Sub BulkInsert(Of T) ( _
    obj As IEnumerable(Of T), _
    tableName As String _
)
```

generic<typename T>  
```
void BulkInsert(  
    IEnumerable<T>^ obj,
    String^ tableName
)
```

**Type Parameters**

T

**Parameters**

obj  
Type: IEnumerable(Of (Of T))

tableName  
Type: String
See Also

IDbStore Interface
SimpleEssentials.DataStore Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
IDbStore..:::.Delete<((Of <('T')>)>) Method

Namespace: SimpleEssentials.DataStore
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#

bool Delete<T>(
    T obj
)

Function Delete(Of T) ( _
    obj As T _
) As Boolean

generic<typename T>

bool^ Delete(
    T obj
)

Type Parameters

T

Parameters

obj
    Type: T
See Also

IDbStore Interface
SimpleEssentials.DataStore Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
IDbStore:::Execute Method

Namespace:  SimpleEssentials.DataStore
Assembly:  SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#
VB
C++

int Execute(
    string sql,
    Object param
)

Function Execute (_
    sql As String, _
    param As Object _
) As Integer

int^ Execute(
    String^ sql,
    Object^ param
)

Parameters

sql
    Type: String

param
    Type: Object
See Also

IDbStore Interface
SimpleEssentials.DataStore Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
IDbStore..::..ExecuteScalar Method

Namespace:  SimpleEssentials.DataStore
Assembly:  SimpleEssentials (in SimpleEssentials.dll)
Syntax

**C#**

```csharp
int ExecuteScalar(
    string sql,
    Object param
)
```

**VB**

```vbnet
Function ExecuteScalar ( _
    sql As String, _
    param As Object _
) As Integer
    int^ ExecuteScalar( _
        String^ sql, _
        Object^ param
    )
```

**C++**

```cpp
int ExecuteScalar(
    string sql,
    Object param
)
```

### Parameters

- **sql**
  - Type: String

- **param**
  - Type: Object
See Also

IDbStore Interface
SimpleEssentials.DataStore Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
IDbStore..::.Get(Of <('T')>)

Method

Namespace: SimpleEssentials.DataStore
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#

VB

C++

T Get<T>(
    Object id
)

Function Get(Of T) ( _
    id As Object _
) As T

generic<typeparam T>
T Get( Object^ id

Type Parameters

T

Parameters

id
    Type: Object
See Also

IDbStore Interface
SimpleEssentials.DataStore Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
IDbStore...:::GetByParameters<(Of <(<'T'>)>)> Method

Namespace: SimpleEssentials.DataStore
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

**C#**

```csharp
IEnumerable<T> GetByParameters<T>(
    string sql,
    Object param
)
```

**VB**

```vbnet
Function GetByParameters(Of T) ( _
    sql As String, _
    param As Object _
) As IEnumerable(Of T)
```

**C++**

```c++
generic<typename T>
IEnumerable<T>^ GetByParameters(
    String^ sql,
    Object^ param
)
```

**Type Parameters**

T

**Parameters**

sql
  Type: **String**

param
  Type: **Object**
See Also

IDbStore Interface
SimpleEssentials.DataStore Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
IDbStore..::..GetByType<(<'T'>)> Method

Namespace: SimpleEssentials.DataStore
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

**C#**

```csharp
IEnumerable<T> GetByType<T>()
```

**VB**

```vbnet
Function GetByType(Of T) As IEnumerable(Of T)
```

generic<typename T>

```vbnet
IEnumerable<T> ^ GetByType()
```

**Type Parameters**

T
See Also

IDbStore Interface
SimpleEssentials.DataStore Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
IDbStore...:::GetMultiMap<(Of
<('T, T2)>)> Method (String,
Func<(Of <('T, T2, T)>)>), Object,
String)

Namespace:  SimpleEssentials.DataStore
Assembly:  SimpleEssentials (in SimpleEssentials.dll)
**Syntax**

**C#**

```csharp
IEnumerable<T> GetMultiMap<T, T2>(
    string sql,
    Func<T, T2, T> func,
    Object param,
    string splitOn
)
```

**VB**

```vbnet
Function GetMultiMap(Of T, T2) (  _
    sql As String,  _
    func As Func(Of T, T2, T), _
    param As Object, _
    splitOn As String  _
) As IEnumerable(Of T)
```

**C++**

```c++
generic<typename T, typename T2>
IEnumerable<T>^ GetMultiMap(
    String^ sql,
    Func<T, T2, T^> func,
    Object^ param,
    String^ splitOn
)
```

**Type Parameters**

- **T**
- **T2**

**Parameters**

- **sql**
  - Type: `String`
- **func**
  - Type: `Func<Of <(<'T, T2, T')>>`
param
  Type: Object

splitOn
  Type: String
See Also

- IDbStore Interface
- GetMultiMap Overload
- SimpleEssentials.DataStore Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
IDbStore...:..GetMultiMap<(Of
<('T, T2, T3>>)>
Method (String,
Func<(Of<('T, T2, T3, T>>)>,
Object, String)

Namespace: SimpleEssentials.DataStore
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#

VB

C++

```csharp
IEnumerable<T> GetMultiMap<T, T2, T3>(
    string sql,
    Func<T, T2, T3, T> func,
    Object param,
    string splitOn
)

Function GetMultiMap(Of T, T2, T3) ( _
    sql As String, _
    func As Func(Of T, T2, T3, T), _
    param As Object, _
    splitOn As String _
) As IEnumerable(Of T)

generic<typename T, typename T2, typename T3>
IEnumerable<T>^ GetMultiMap(
    String^ sql,
    Func<T, T2, T3, T>^ func,
    Object^ param,
    String^ splitOn
)
```

**Type Parameters**

T
T2
T3

**Parameters**

sql
    Type: String

func
Type: Func<(Of <('T, T2, T3, T)>)>  

param  
   Type: Object  

splitOn  
   Type: String
See Also

- IDbStore Interface
- GetMultiMap Overload
- SimpleEssentials.DataStore Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
IDbStore...:::GetMultiMap<(Of <('T, T2, T3, T4>>)>) Method (String, Func<(Of <('T, T2, T3, T4, T>>)>>, Object, String)

Namespace: SimpleEssentials.DataStore
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#
VB
C++

**IEnumerable**<T> GetMultiMap<T, T2, T3, T4>(
    string sql,
    Func<T, T2, T3, T4, T> func,
    Object param,
    string splitOn
)

Function GetMultiMap(Of T, T2, T3, T4) ( _
    sql As String, _
    func As Func(Of T, T2, T3, T4, T), _
    param As Object, _
    splitOn As String _
) As IEnumerable(Of T)

generic<typename T, typename T2, typename T3, typename T4>
**IEnumerable**<T>^ GetMultiMap(
    String^ sql,
    Func<T, T2, T3, T4, T>^ func,
    Object^ param,
    String^ splitOn
)

**Type Parameters**

T
T2
T3
T4

**Parameters**

sql
    Type: String
func
    Type: Func<((<T, T2, T3, T4, T>)>)

param
    Type: Object

splitOn
    Type: String
See Also

IDbStore Interface
GetMultiMap Overload
SimpleEssentials.DataStore Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
IDbStore::GetMultiMap<Of <('T, T2, T3, T4, T5)>)> Method (String, Func<Of <('T, T2, T3, T4, T5, T)>>, Object, String)

Namespace: SimpleEssentials.DataStore
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#

```csharp
IEnumerable<T> GetMultiMap<T, T2, T3, T4, T5>(
    string sql,
    Func<T, T2, T3, T4, T5, T> func,
    Object param,
    string splitOn
)
```

VB

```vb
Function GetMultiMap(Of T, T2, T3, T4, T5) (_
    sql As String, _
    func As Func(Of T, T2, T3, T4, T5, T), _
    param As Object, _
    splitOn As String _
) As IEnumerable(Of T)
```

generic<typename T, typename T2, typename T3, typename T4, typename T5>

```csharp
IEnumerable<T>^ GetMultiMap(
    String^ sql,
    Func<T, T2, T3, T4, T5, T>^ func,
    Object^ param,
    String^ splitOn
)
```

Type Parameters

- **T**
- **T2**
- **T3**
- **T4**
- **T5**

Parameters

- **sql**
  - Type: **String**
func
  Type: Func<((T, T2, T3, T4, T5, T)>)>

param
  Type: Object

splitOn
  Type: String
See Also

IDbStore Interface
GetMultiMap Overload
SimpleEssentials.DataStore Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
IDbStore...:::GetMultiMap<Of
<('T, T2, T3, T4, T5, T6)>)>
Method (String, Func<Of <('T, T2, T3, T4, T5, T6, T)>)>, Object, String)

Namespace: SimpleEssentials.DataStore
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#
VB
C++

IEnumerable<T> GetMultiMap<T, T2, T3, T4, T5, T6>(
    string sql,
    Func<T, T2, T3, T4, T5, T6, T> func,
    Object param,
    string splitOn
)

Function GetMultiMap(Of T, T2, T3, T4, T5, T6) ( _
    sql As String, _
    func As Func(Of T, T2, T3, T4, T5, T6, T), _
    param As Object, _
    splitOn As String _
) As IEnumerable(Of T)

generic<syscall T, syscall T2, syscall T3, syscall T4, syscall T5, syscall T6>
IEnumerable<T> GetMultiMap(
    String^ sql,
    Func<T, T2, T3, T4, T5, T6, T>^ func,
    Object^ param,
    String^ splitOn
)

Type Parameters

T
T2
T3
T4
T5
T6

Parameters

sql
Type: **String**

func
  Type: Func<(<T, T2, T3, T4, T5, T6, T>)>

param
  Type: **Object**

splitOn
  Type: **String**
See Also

IDbStore Interface
GetMultiMap Overload
SimpleEssentials.DataStore Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
IDbStore..::..Update(Of 'T)>

Method

Namespace: SimpleEssentials.DataStore
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#

```csharp
bool Update<T>(
    T obj
)
```

VB

```vbnet
Function Update(Of T) ( _
    obj As T _
) As Boolean
```

eraldg<typename T>

```csharp
bool^ Update(
    T obj
)
```

Type Parameters

T

Parameters

obj

Type: T
See Also

IDbStore Interface
SimpleEssentials.DataStore Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
SimpleEssentials.Diagnostics Namespace
## Classes

<table>
<thead>
<tr>
<th>Class</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Log</td>
<td></td>
</tr>
<tr>
<td>LogFileByDateHandler</td>
<td></td>
</tr>
</tbody>
</table>
## Interfaces

<table>
<thead>
<tr>
<th>Interface</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ILog</td>
<td></td>
</tr>
<tr>
<td>ILogFileHandler</td>
<td></td>
</tr>
</tbody>
</table>

*Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.*
**ILog Interface**

**Namespace:**  SimpleEssentials.Diagnostics

**Assembly:**  SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#
VB
C++

public interface ILog

Public Interface ILog

public interface class ILog
See Also

ILog Members
SimpleEssentials.Diagnostics Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
ILog Members

The ILog type exposes the following members.
# Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Debug</strong></td>
<td>Determines whether the specified object is equal to the current object.</td>
</tr>
<tr>
<td><strong>Equals(System.Object)</strong></td>
<td>(Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td><strong>Error(String)</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Error(String, Exception)</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Finalize</strong></td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Serves as the default hash function. (Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td>Gets the <a href="#">Type</a> of the current instance. (Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td><strong>Info</strong></td>
<td></td>
</tr>
<tr>
<td><strong>MemberwiseClone</strong></td>
<td>Creates a shallow copy of the current <a href="#">Object</a>. (Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td>Returns a string that represents the current object. (Inherited from <a href="#">Object</a>.)</td>
</tr>
</tbody>
</table>
See Also

ILog Interface
SimpleEssentials.Diagnostics Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
**ILog Methods**

The `ILog` type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Debug</td>
<td>Determines whether the specified object is equal to the current object.</td>
</tr>
<tr>
<td>error(String)</td>
<td>Error(String, Exception)</td>
</tr>
<tr>
<td>JsonResult</td>
<td>Finalize</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection.</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function.</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance.</td>
</tr>
<tr>
<td>Info</td>
<td>MemberwiseClone</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object.</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object.</td>
</tr>
</tbody>
</table>
See Also

ILog Interface
SimpleEssentials.Diagnostics Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
ILog::<..::Debug Method

Namespace: SimpleEssentials.Diagnostics
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#

void Debug(
    string msg
)

VB

Sub Debug ( _
    msg As String _
)

C++

void Debug(
    string^ msg
)

Parameters

msg
    Type: String
See Also

ILog Interface
SimpleEssentials.Diagnostics Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
ILog:::..Error Method
# Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>⚠️ Error(String)</td>
<td></td>
</tr>
<tr>
<td>⚠️ Error(String, Exception)</td>
<td></td>
</tr>
</tbody>
</table>
See Also

ILog Interface
ILog Members
SimpleEssentials.Diagnostics Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
ILog..::..Error Method (String)

Namespace: SimpleEssentials.Diagnostics
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#

```csharp
void Error(
    string msg
)
```

VB

```vbnet
Sub Error (_
    msg As String _
)
```

C++

```cpp
void Error(
    String^ msg
)
```

Parameters

msg
  Type: String
See Also

ILog Interface  
Error Overload  
SimpleEssentials.Diagnostics Namespace  

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
ILog..::..Error Method (String, Exception)

Namespace: SimpleEssentials.Diagnostics
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#
VB
C++

void Error(
    string msg,
    Exception exception
)

Sub Error ( _
    msg As String, _
    exception As Exception _
)

void Error(
    String^ msg,
    Exception^ exception
)

Parameters

msg
    Type: String

exception
    Type: Exception
See Also

ILog Interface
Error Overload
SimpleEssentials.Diagnostics Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
ILog:::..Info Method

Namespace: SimpleEssentials.Diagnostics
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#

void Info(
    string msg
)

Sub Info (_
    msg As String _
)

void Info(_
    String^ msg
)

Parameters

msg
    Type: String
See Also

ILog Interface
SimpleEssentials.Diagnostics Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
ILogFileHandler Interface

Namespace: SimpleEssentials.Diagnostics
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#
VB
C++

public interface ILogFileHandler

Public Interface ILogFileHandler

public interface class ILogFileHandler
See Also

ILogFileHandler Members
SimpleEssentials.Diagnostics Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
ILogFileHandler Members

The ILogFileHandler type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Equals(System.Object)</code></td>
<td>Determines whether the specified object is equal to the current object. (Inherited from <code>Object</code>)</td>
</tr>
<tr>
<td><code>Finalize</code></td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from <code>Object</code>)</td>
</tr>
<tr>
<td><code>GetHashCode</code></td>
<td>Serves as the default hash function. (Inherited from <code>Object</code>)</td>
</tr>
<tr>
<td><code>GetType</code></td>
<td>Gets the <code>Type</code> of the current instance. (Inherited from <code>Object</code>)</td>
</tr>
<tr>
<td><code>Insert</code></td>
<td></td>
</tr>
<tr>
<td><code>MemberwiseClone</code></td>
<td>Creates a shallow copy of the current <code>Object</code>. (Inherited from <code>Object</code>)</td>
</tr>
<tr>
<td><code>ToString</code></td>
<td>Returns a string that represents the current object. (Inherited from <code>Object</code>)</td>
</tr>
</tbody>
</table>
See Also

ILogFileHandler Interface
SimpleEssentials.Diagnostics Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
**ILogFileHandler Methods**

The `ILogFileHandler` type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Equals(System.Object)</strong></td>
<td>Determines whether the specified object is equal to the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td><strong>Finalize</strong></td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td><strong>Insert</strong></td>
<td></td>
</tr>
<tr>
<td><strong>MemberwiseClone</strong></td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td>Returns a string that represents the current object. (Inherited from Object.)</td>
</tr>
</tbody>
</table>
See Also

ILogFileHandler Interface
SimpleEssentials.Diagnostics Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
ILogFileHandler:::..Insert Method

Namespace: SimpleEssentials.Diagnostics
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#

```c#
void Insert(
    string msg
)
```

VB

```vb
Sub Insert (_
    msg As String _
)
```

C++

```cpp
void Insert( 
    String^ msg
)
```

Parameters

msg

Type: String
See Also

ILogFileHandler Interface
SimpleEssentials.Diagnostics Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
Log Class

Namespace:  SimpleEssentials.Diagnostics
Assembly:  SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#

public class Log : ILog

VB

Public Class Log
    Implements ILog

C++

public ref class Log : ILog
Inheritance Hierarchy

Object
SimpleEssentials.Diagnostics...Log
See Also

Log Members
SimpleEssentials.Diagnostics Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
Log Members

The Log type exposes the following members.
# Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Debug</td>
<td>Determines whether the specified object is equal to the current object.</td>
</tr>
<tr>
<td>Equals(System.Object)</td>
<td>(Inherited from Object.)</td>
</tr>
<tr>
<td>Error(String)</td>
<td></td>
</tr>
<tr>
<td>Error(String, Exception)</td>
<td></td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>Info</td>
<td></td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from Object.)</td>
</tr>
</tbody>
</table>
See Also

Log Class
SimpleEssentials.Diagnostics Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
Log Methods

The Log type exposes the following members.
# Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Debug</code></td>
<td>Determines whether the specified object is equal to the current object.</td>
</tr>
<tr>
<td><code>Equals(System.Object)</code></td>
<td>(Inherited from <code>Object</code>.</td>
</tr>
<tr>
<td><code>Error(String)</code></td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection.</td>
</tr>
<tr>
<td><code>Error(String, Exception)</code></td>
<td>(Inherited from <code>Object</code>.</td>
</tr>
<tr>
<td><code>Finalize</code></td>
<td>Serves as the default hash function.</td>
</tr>
<tr>
<td><code>GetHashCode</code></td>
<td>(Inherited from <code>Object</code>.</td>
</tr>
<tr>
<td><code>GetType</code></td>
<td>Gets the <code>Type</code> of the current instance.</td>
</tr>
<tr>
<td><code>Info</code></td>
<td>(Inherited from <code>Object</code>.</td>
</tr>
<tr>
<td><code>MemberwiseClone</code></td>
<td>Creates a shallow copy of the current <code>Object</code>.</td>
</tr>
<tr>
<td><code>ToString</code></td>
<td>(Inherited from <code>Object</code>.</td>
</tr>
</tbody>
</table>
See Also

Log Class
SimpleEssentials.Diagnostics Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
Log...:::Debug Method

**Namespace:**  SimpleEssentials.Diagnostics  
**Assembly:**  SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#

VB

C++

public void Debug(
    string msg
)

Public Sub Debug (_
    msg As String _
)

public:
void Debug(
    String^ msg
)

Parameters

msg
    Type: String
See Also

Log Class
SimpleEssentials.Diagnostics Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
Log...::..Error Method
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Error(String)</code></td>
<td></td>
</tr>
<tr>
<td><code>Error(String, Exception)</code></td>
<td></td>
</tr>
</tbody>
</table>
See Also

Log Class
Log Members
SimpleEssentials.Diagnostics Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
Log:::..Error Method (String)

Namespace: SimpleEssentials.Diagnostics
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#
VB
C++

public void Error(
    string msg
)

Public Sub Error ( _
    msg As String _
)

public:
void Error(
    String^ msg
)

Parameters

msg
    Type: String
See Also

Log Class
Error Overload
SimpleEssentials.Diagnostics Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
Log:::Error Method (String, Exception)

Namespace: SimpleEssentials.Diagnostics
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#
VB
C++

public void Error(  
    string msg,  
    Exception exception
)

Public Sub Error ( _  
   msg As String, _  
   exception As Exception _
)

public:
void Error(  
    String^ msg,  
    Exception^ exception
)

Parameters

msg
   Type: String

exception
   Type: Exception
See Also

Log Class
Error Overload
SimpleEssentials.Diagnostics Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
Log:::..Info Method

Namespace: SimpleEssentials.Diagnostics
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#

public void Info(
    string msg
)

VB

Public Sub Info ( _
    msg As String _
)

C++

public:
void Info(
    String^ msg
)

Parameters

msg
    Type: String
See Also

Log Class
SimpleEssentials.Diagnostics Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
LogFileByDateHandler Class

Namespace:  SimpleEssentials.Diagnostics
Assembly:  SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#  
Public class LogFileByDateHandler : ILogFileHandler

VB  
Public Class LogFileByDateHandler  
    Implements ILogFileHandler

C++  
public ref class LogFileByDateHandler : ILogFileHandler
Inheritance Hierarchy

Object
SimpleEssentials.Diagnostics...LogFileByDateHandler
See Also

LogFileByDateHandler Members
SimpleEssentials.Diagnostics Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
LogFileByDateHandler Members

The LogFileByDateHandler type exposes the following members.
# Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LogFileByDateHandler</td>
<td></td>
</tr>
</tbody>
</table>
# Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="Equals(System.Object)" alt="" /></td>
<td>Determines whether the specified object is equal to the current object.  (Inherited from Object.)</td>
</tr>
<tr>
<td><img src="Finalize" alt="" /></td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)</td>
</tr>
<tr>
<td><img src="GetHashCode" alt="" /></td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td><img src="GetType" alt="" /></td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td><img src="Insert" alt="" /></td>
<td></td>
</tr>
<tr>
<td><img src="MemberwiseClone" alt="" /></td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td><img src="ToString" alt="" /></td>
<td>Returns a string that represents the current object. (Inherited from Object.)</td>
</tr>
</tbody>
</table>
See Also

LogFileByDateHandler Class
SimpleEssentials.Diagnostics Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
LogFileByDateHandler Constructor

Namespace:  SimpleEssentials.Diagnostics
Assembly:  SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#
VB
C++

public LogFileByDateHandler()

Public Sub New

public:
LogFileByDateHandler()
See Also

LogFileByDateHandler Class
SimpleEssentials.Diagnostics Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
LogFileByDateHandler Methods

The LogFileByDateHandler type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Equals(System.Object)</code></td>
<td>Determines whether the specified object is equal to the current object.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td><code>Finalize</code></td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td><code>GetHashCode</code></td>
<td>Serves as the default hash function.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td><code>GetType</code></td>
<td>Gets the <a href="#">Type</a> of the current instance.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td><code>Insert</code></td>
<td></td>
</tr>
<tr>
<td><code>MemberwiseClone</code></td>
<td>Creates a shallow copy of the current <a href="#">Object</a>.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td><code>ToString</code></td>
<td>Returns a string that represents the current object.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">Object</a>.)</td>
</tr>
</tbody>
</table>
See Also

LogFileByDateHandler Class
SimpleEssentials.Diagnostics Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
LogFileByDateHandler::<..::Insert Method

Namespace: SimpleEssentials.Diagnostics
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#
public void Insert(
    string msg
)

VB
Public Sub Insert ( _
    msg As String _
)

C++
public:
void Insert(  
    String^ msg
)

Parameters

msg
    Type: String
See Also

LogFileByDateHandler Class
SimpleEssentials.Diagnostics Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
SimpleEssentials.Extensions
Namespace
# Classes

<table>
<thead>
<tr>
<th>Class</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>🎊 ConsoleEx</td>
<td>Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.</td>
</tr>
</tbody>
</table>
ConsoleEx Class

Namespace:  SimpleEssentials.Extensions
Assembly:  SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#

Public static class ConsoleEx

VB

Public NotInheritable Class ConsoleEx

C++

public ref class ConsoleEx abstract sealed
Inheritance Hierarchy

Object
SimpleEssentials.Extensions.....ConsoleEx
See Also

ConsoleEx Members
SimpleEssentials.Extensions Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
ConsoleEx Members

The ConsoleEx type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>❯ WriteLine</td>
<td></td>
</tr>
<tr>
<td>❯ WriteTitle</td>
<td></td>
</tr>
</tbody>
</table>
See Also

ConsoleEx Class
SimpleEssentials.Extensions Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
ConsoleEx Methods

The ConsoleEx type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>WriteLine</td>
<td></td>
</tr>
<tr>
<td>WriteTitle</td>
<td></td>
</tr>
</tbody>
</table>
See Also

ConsoleEx Class
SimpleEssentials.Extensions Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
ConsoleEx::WriteLine Method

Namespace:  SimpleEssentials.Extensions
Assembly:  SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#

```csharp
public static void WriteLine(
    string text,
    ConsoleColor color
)
```

VB

```vb
Public Shared Sub WriteLine ( _
    text As String, _
    color As ConsoleColor _
)
```

C++

```cpp
public:
static void WriteLine(
    String^ text,
    ConsoleColor^ color
)
```

Parameters

text
   Type: `String`

color
   Type: `ConsoleColor`
See Also

ConsoleEx Class
SimpleEssentials.Extensions Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
ConsoleEx...WriteTitle Method

Namespace: SimpleEssentials.Extensions
Assembly: SimpleEssentials (in SimpleEssentials.dll)
## Syntax

**C#**

```csharp
public static void WriteTitle(
    string text,
    int tileCount,
    ConsoleColor color
)
```

**VB**

```vbnet
Public Shared Sub WriteTitle ( _
    text As String, _
    tileCount As Integer, _
    color As ConsoleColor _
)
```

**C++**

```c++
public:
static void WriteTitle( _
    String^ text,
    int^ tileCount,
    ConsoleColor^ color)
```

### Parameters

- **text**
  - Type: `String`

- **tileCount**
  - Type: `Int32`

- **color**
  - Type: `ConsoleColor`
See Also

ConsoleEx Class
SimpleEssentials.Extensions Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
SimpleEssentials.IO Namespace
## Classes

<table>
<thead>
<tr>
<th>Class</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FileHandler</td>
<td></td>
</tr>
<tr>
<td>FolderHandler</td>
<td></td>
</tr>
</tbody>
</table>
## Interfaces

<table>
<thead>
<tr>
<th>Interface</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IFileHandler</td>
<td></td>
</tr>
<tr>
<td>IFolderHandler</td>
<td></td>
</tr>
<tr>
<td>IHandler</td>
<td></td>
</tr>
</tbody>
</table>

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
FileHandler Class

Namespace:  SimpleEssentials.IO
Assembly:  SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#
VB
C++

public class FileHandler : IFileHandler

Public Class FileHandler Implements IFileHandler

public ref class FileHandler : IFileHandler
Inheritance Hierarchy

Object
SimpleEssentials.IO...FileHandler
See Also

FileHandler Members
SimpleEssentials.IO Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
FileHandler Members

The FileHandler type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create(String)</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>Create(String, IFolder)</td>
<td></td>
</tr>
<tr>
<td>Equals(System.Object)</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>Get</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Returns a string that represents the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td></td>
</tr>
<tr>
<td>Move</td>
<td></td>
</tr>
<tr>
<td>Read(IFile)</td>
<td></td>
</tr>
<tr>
<td>Read&lt;(Of &lt;&lt;(T)&gt;&gt;)(IFile, IFileReader)</td>
<td></td>
</tr>
<tr>
<td>ReadAll&lt;(Of &lt;&lt;(T)&gt;&gt;)</td>
<td></td>
</tr>
<tr>
<td>ReadBy&lt;(Of &lt;&lt;(T)&gt;&gt;)</td>
<td></td>
</tr>
<tr>
<td>Rename</td>
<td></td>
</tr>
<tr>
<td>ToString</td>
<td></td>
</tr>
<tr>
<td>Write(IFile, String, Boolean)</td>
<td></td>
</tr>
<tr>
<td>Write&lt;(Of &lt;&lt;(T)&gt;&gt;)(IFile, T, IFileWriter, Boolean)</td>
<td></td>
</tr>
<tr>
<td>Write&lt;(Of &lt;&lt;(T)&gt;&gt;)(IFile,</td>
<td></td>
</tr>
</tbody>
</table>
IEnumerable(Of IFileWriter, Boolean)
See Also

FileHandler Class
SimpleEssentials.IO Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
FileHandler Methods

The FileHandler type exposes the following members.
# Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>(Inherited from)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create(String)</td>
<td>Determines whether the specified object is equal to the current object.</td>
<td>Object.</td>
</tr>
<tr>
<td>Create(String, IFolder)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Equals(System.Object)</td>
<td>Determines whether the specified object is equal to the current object.</td>
<td>Object.</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection.</td>
<td>Object.</td>
</tr>
<tr>
<td>Get</td>
<td></td>
<td></td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function.</td>
<td>Object.</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the <code>Type</code> of the current instance.</td>
<td>Object.</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current <code>Object</code>.</td>
<td>Object.</td>
</tr>
<tr>
<td>Move</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Read(IFile)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Read&lt;(Of &lt;&lt;(T)&gt;&gt;)&gt;(IFile, IFileReader)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ReadAll&lt;(Of &lt;&lt;'(T)&gt;&gt;)&gt;)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ReadBy&lt;(Of &lt;&lt;'(T)&gt;&gt;)&gt;)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Rename</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object.</td>
<td>Object.</td>
</tr>
<tr>
<td>Write(IFile, String, Boolean)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Write&lt;(Of &lt;&lt;'(T)&gt;&gt;)&gt;(IFile, T, IFileWriter, Boolean)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Write&lt;(Of &lt;&lt;'(T)&gt;&gt;)&gt;(IFile,</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
IEnumerable(Of (Of (T))),
IFileWriter, Boolean)
See Also

**FileHandler Class**
**SimpleEssentials.IO Namespace**

*Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.*
FileHandler...Create Method
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create(String)</td>
<td></td>
</tr>
<tr>
<td>Create(String, IFolder)</td>
<td></td>
</tr>
</tbody>
</table>
See Also

FileHandler Class
FileHandler Members
SimpleEssentials.IO Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
FileHandler...Create Method
(String)

Namespace: SimpleEssentials.IO
Assembly: SimpleEssentials (in SimpleEssentials.dll)
### Syntax

**C#**
```csharp
public IFileType Create(
    string path
)
```

**VB**
```vbnet
Public Function Create (_
    path As String _
) As IFileType
```

**C++**
```cpp
public:
IFileType^ Create(
    String^ path
)
```

**Parameters**

`path`
Type: `String`
See Also

FileHandler Class
Create Overload
SimpleEssentials.IO Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
FileHandler..::..Create Method
(String, IFolder)

Namespace:  SimpleEssentials.IO
Assembly:  SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#
VB
C++

public IFile Create(
    string fileName,
    IFolder parentFolder
)

Public Function Create ( _
    fileName As String, _
    parentFolder As IFolder _
) As IFile

public:
IFile^ Create(
    String^ fileName,
    IFolder^ parentFolder
)

Parameters

fileName
    Type: String

parentFolder
    Type: SimpleEssentials.IO.Types...IFolder
See Also

FileHandler Class
Create Overload
SimpleEssentials.IO Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
FileHandler...Get Method

Namespace:  SimpleEssentials.IO
Assembly:  SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#

VB

C++

public IFileType Get(
  string path
)

Public Function Get (_
  path As String _
) As IFileType

public:
IFileType^ Get(
  String^ path
)

Parameters

path
  Type: String
See Also

FileHandler Class
SimpleEssentials.IO Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
FileHandler...Move Method

Namespace: SimpleEssentials.IO
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#

```csharp
public bool Move(
    ref IFileType file,
    string newPath
)
```

VB

```vb
Public Function Move (
    ByRef file As IFileType,
    newPath As String
) As Boolean
    public:
    bool^ Move(
        IFileType^ file,
        String^ newPath
    )
```

Parameters

file
  Type: SimpleEssentials.IO.Types..::..IFileType%

newPath
  Type: String
See Also

FileHandler Class
SimpleEssentials.IO Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
FileHandler...:::..Read Method (IFile)

Namespace: SimpleEssentials.IO
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#  
VB  
C++

public string Read(    
    IFile file
)    

Public Function Read ( _    
    file As IFile _
) As String

public:    
String^ Read(    
    IFile^ file
)

Parameters

file    
Type: SimpleEssentials.IO.Types::: IFile
See Also

FileHandler Class
Read Overload
SimpleEssentials.IO Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
FileHandler..::..Read(Of (<'T'>))> Method (IFile, IFileReader)

Namespace: SimpleEssentials.IO
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

**C#**

```csharp
public T Read<T>(
    IFile file,
    IFileReader fileReader
)
```

**VB**

```vbnet
Public Function Read(Of T) ( _
    file As IFile, _
    fileReader As IFileReader _
) As T
```

**C++**

```cpp
public:
    template<typename T>
    T Read( _
        IFile file,
        IFileReader fileReader
    )
```

**Type Parameters**

- **T**

**Parameters**

- **file**
  Type: `SimpleEssentials.IO.Types.IFile`

- **fileReader**
  Type: `SimpleEssentials.IO.Readers.IFileReader`
See Also

FileHandler Class
Read Overload
SimpleEssentials.IO Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
FileHandler...ReadAll<Of<('T>)>> Method

Namespace: SimpleEssentials.IO
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#
VB
C++

public IEnumerable<T> ReadAll<T>(
    IFile file,
    IFileReader fileReader
)

Public Function ReadAll(Of T) ( _
    file As IFile, _
    fileReader As IFileReader _
) As IEnumerable(Of T)

public:
generic<typename T>
IEnumerable<T>^ ReadAll(
    IFile^ file,
    IFileReader^ fileReader
)

Type Parameters

T

Parameters

file
    Type: SimpleEssentials.IO.Types.IFile

fileReader
    Type: SimpleEssentials.IO.Readers.IFileReader
See Also

FileHandler Class
SimpleEssentials.IO Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
FileHandler...:::..ReadBy<(Of <('T')>)> Method

Namespace: SimpleEssentials.IO
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#

```csharp
public IEnumerable<T> ReadBy<T>(
    IFile file,
    Func<T, bool> predicate,
    IFileReader fileReader
)
```

VB

```vbnet
Public Function ReadBy(Of T) ( _
    file As IFile, _
    predicate As Func(Of T, Boolean), _
    fileReader As IFileReader _
) As IEnumerable(Of T)
```

C++

```cpp
public:

generic<

typename T>

IEnumerable<T>n^ ReadBy(
    IFile^ file,
    Func<T, bool^n>^ predicate,
    IFileReader^ fileReader
)
```

Type Parameters

T

Parameters

file

Type: SimpleEssentials.IO.Types..::.IFile

predicate

Type: Func<(Of <'T, Boolean>)>)

fileReader

Type: SimpleEssentials.IO.Readers..::.IFileReader
See Also

FileHandler Class
SimpleEssentials.IO Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
FileHandler...::..Rename Method

Namespace:  SimpleEssentials.IO
Assembly:  SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#  
public bool Rename(
    ref IFileType file,
    string newName
)
See Also

FileHandler Class
SimpleEssentials.IO Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
FileHandler...Write Method
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Write(IFile, String, Boolean)</td>
<td></td>
</tr>
<tr>
<td>Write(Of &lt;&lt;'T'&gt;&gt;)&gt;(IFile, T, IFileWriter, Boolean)</td>
<td></td>
</tr>
<tr>
<td>Write(Of &lt;&lt;'T'&gt;&gt;)&gt;(IFile, IEnumerable(Of &lt;&lt;'T'&gt;&gt;, IFileWriter, Boolean)</td>
<td></td>
</tr>
</tbody>
</table>
See Also

FileHandler Class
FileHandler Members
SimpleEssentials.IO Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
FileHandler..::.Write Method (IFile, String, Boolean)

Namespace:  SimpleEssentials.IO
Assembly:  SimpleEssentials (in SimpleEssentials.dll)
Syntax

**C#**

```csharp
public bool Write(
    IFile file,
    string content,
    bool append)
```

**VB**

```vbnet
Public Function Write ( _
    file As IFile, _
    content As String, _
    append As Boolean _
) As Boolean

public:
    bool Write(
        IFile^ file, 
        String^ content, 
        bool^ append 
    )
```

**Parameters**

- **file**
  - Type: `SimpleEssentials.IO.Types.IFile`

- **content**
  - Type: `String`

- **append**
  - Type: `Boolean`
See Also

FileHandler Class
Write Overload
SimpleEssentials.IO Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
FileHandler..::.Write(Of (\'<T\>')>) Method (IFile, T, IFileWriter, Boolean)

Namespace: SimpleEssentials.IO
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#
VB
C++

public void Write<T>(
    IFile file,
    T obj,
    IFileWriter fileWriter,
    bool append
)

Public Sub Write(Of T) ( _
    file As IFile, _
    obj As T, _
    fileWriter As IFileWriter, _
    append As Boolean _
)

public:
    generic<typename T>
    void Write(
        IFile^ file,
        T obj,
        IFileWriter^ fileWriter,
        bool^ append
    )

Type Parameters

T

Parameters

file
    Type: SimpleEssentials.IO.Types.;.;.;IFile

obj
    Type: T
fileWriter
    Type: SimpleEssentials.IO.Writers.IFileWriter

append
    Type: Boolean
See Also

FileHandler Class
Write Overload
SimpleEssentials.IO Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
FileHandler...:::Write<(Of <('T'>)>)> Method (IFile, IEnumerable<(Of <('T'>)>)>, IFileWriter, Boolean)

Namespace: SimpleEssentials.IO
Assembly: SimpleEssentials (in SimpleEssentials.dll)
### Syntax

**C#**
```
public void Write<T>(
    IFile file,
    IEnumerable<T> obj,
    IFileWriter fileWriter,
    bool append
)
```

**VB**
```
Public Sub Write(Of T) ( _
    file As IFile, _
    obj As IEnumerable(Of T), _
    fileWriter As IFileWriter, _
    append As Boolean _
)
```

**C++**
```
public:
    template<typename T>
    void Write( 
        IFile file,
        IEnumerable<T> obj,
        IFileWriter fileWriter,
        bool append
    )
```

**Type Parameters**

**T**

**Parameters**

**file**
Type: `SimpleEssentials.IO.Types::IFile`

**obj**
Type: `IEnumerable<Of (Of 'T)>`
fileWriter
  Type: SimpleEssentials.IO.Writers..,..,IFileWriter

append
  Type: Boolean
See Also

FileHandler Class
Write Overload
SimpleEssentials.IO Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
FolderHandler Class

Namespace: SimpleEssentials.IO
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#
VB
C++

public class FolderHandler : IFolderHandler

Public Class FolderHandler __
    Implements IFolderHandler

public ref class FolderHandler : IFolderHandler
Inheritance Hierarchy

Object
SimpleEssentials.IO...FolderHandler
See Also

FolderHandler Members
SimpleEssentials.IO Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
FolderHandler Members

The FolderHandler type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create(String)</td>
<td>Determines whether the specified object is equal to the current object.</td>
</tr>
<tr>
<td>Create(String, Boolean)</td>
<td>(Inherited from Object.)</td>
</tr>
<tr>
<td>Create(String, IFolder)</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection.</td>
</tr>
<tr>
<td>Equals(System.Object)</td>
<td>(Inherited from Object.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Serves as the default hash function.</td>
</tr>
<tr>
<td>Get</td>
<td>(Inherited from Object.)</td>
</tr>
<tr>
<td>GetChildFiles</td>
<td>Gets the Type of the current instance.</td>
</tr>
<tr>
<td>GetChildFolders</td>
<td>(Inherited from Object.)</td>
</tr>
<tr>
<td>GetChildren</td>
<td>Creates a shallow copy of the current Object.</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>(Inherited from Object.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Returns a string that represents the current object.</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>(Inherited from Object.)</td>
</tr>
<tr>
<td>Move</td>
<td>(Inherited from Object.)</td>
</tr>
<tr>
<td>Rename</td>
<td>(Inherited from Object.)</td>
</tr>
<tr>
<td>ToString</td>
<td>(Inherited from Object.</td>
</tr>
</tbody>
</table>
See Also

FolderHandler Class
SimpleEssentials.IO Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
FolderHandler Methods

The FolderHandler type exposes the following members.
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create(String)</td>
<td></td>
</tr>
<tr>
<td>Create(String, Boolean)</td>
<td></td>
</tr>
<tr>
<td>Create(String, IFolder)</td>
<td></td>
</tr>
<tr>
<td>Equals(System.Object)</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)</td>
</tr>
<tr>
<td>Get</td>
<td></td>
</tr>
<tr>
<td>GetChildFiles</td>
<td></td>
</tr>
<tr>
<td>GetChildFolders</td>
<td></td>
</tr>
<tr>
<td>GetChildren</td>
<td></td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>Move</td>
<td></td>
</tr>
<tr>
<td>Rename</td>
<td></td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from Object.)</td>
</tr>
</tbody>
</table>
See Also

FolderHandler Class
SimpleEssentials.IO Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
FolderHandler...Create Method
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create(String)</td>
<td></td>
</tr>
<tr>
<td>Create(String, Boolean)</td>
<td></td>
</tr>
<tr>
<td>Create(String, JFolder)</td>
<td></td>
</tr>
</tbody>
</table>
See Also

FolderHandler Class
FolderHandler Members
SimpleEssentials.IO Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
FolderHandler...Create Method (String)

Namespace:  SimpleEssentials.IO
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#

public IFileType Create(string path)

VB

Public Function Create(_
    path As String _) As IFileType

C++

public:
IFileType^ Create(_
    String^ path

Parameters

path
    Type: String
See Also

FolderHandler Class
Create Overload
SimpleEssentials.IO Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
FolderHandler...Create Method (String, Boolean)

Namespace: SimpleEssentials.IO
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#

```csharp
public IFolder Create(
    string path,
    bool relative
)
```

VB

```vbnet
Public Function Create (_
    path As String, _
    relative As Boolean _
) As IFolder
```

C++

```cpp
public:
 IFolder^ Create(
    String^ path,
    bool^ relative
)
```

Parameters

path
   Type: String

relative
   Type: Boolean
See Also

FolderHandler Class
Create Overload
SimpleEssentials.IO Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
FolderHandler...Create Method
(String, IFolder)

Namespace: SimpleEssentials.IO
Assembly: SimpleEssentials (in SimpleEssentials.dll)
**Syntax**

**C#**

```
public IFolder Create(
    string path,
    IFolder parent
)
```

**VB**

```
Public Function Create (_
    path As String, _
    parent As IFolder _
) As IFolder
```

**C++**

```
public:
IFolder^ Create(
    String^ path,
    IFolder^ parent
)
```

**Parameters**

- `path`
  - Type: `String`

- `parent`
  - Type: `SimpleEssentials.IO.Types::IFolder`
See Also

FolderHandler Class
Create Overload
SimpleEssentials.IO Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
FolderHandler...Get Method

Namespace: SimpleEssentials.IO
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

**C#**

```csharp
public IFileType Get(
    string path
)
```

**VB**

```vbnet
Public Function Get(
    path As String
) As IFileType
```

**C++**

```cpp
public:
IFileType Get(
    String^ path
)
```

**Parameters**

- **path**
  
  Type: **String**
See Also

FolderHandler Class
SimpleEssentials.IO Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
FolderHandler..::.GetChildFiles

Method

Namespace:  SimpleEssentials.IO
Assembly:  SimpleEssentials (in SimpleEssentials.dll)
Syntax

**C#**
```csharp
public IEnumerable<IFile> GetChildFiles(IFolder parentFolder)
```

**VB**
```vbnet
Public Function GetChildFiles(_
    parentFolder As IFolder _) _
As IEnumerable(Of IFile)
```

**C++**
```cpp
public:
IEnumerable<IFile>^ GetChildFiles(
    IFolder^ parentFolder
)
```

**Parameters**

parentFolder
  Type: SimpleEssentials.IO.Types..::.IFolder
See Also

FolderHandler Class
SimpleEssentials.IO Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
FolderHandler..::.GetChildFolders Method

Namespace: SimpleEssentials.IO
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#
VB
C++

```csharp
public IEnumerable<IFolder> GetChildFolders(
    IFolder parentFolder
)
```

```vbnet
Public Function GetChildFolders ( _
    parentFolder As IFolder _
) As IEnumerable(Of IFolder)
```

```cpp
public:
IEnumerable<IFolder>^ GetChildFolders(
    IFolder^ parentFolder
)
```

Parameters

parentFolder
Type: SimpleEssentials.IO.Types..::..IFolder
See Also

FolderHandler Class
SimpleEssentials.IO Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
FolderHandler...GetChildren Method

Namespace: SimpleEssentials.IO
Assembly: SimpleEssentials (in SimpleEssentials.dll)
## Syntax

**C#**

```csharp
public IEnumerable<IFileType> GetChildren(IFolder parentFolder)
```

**VB**

```vbnet
Public Function GetChildren(_
parentFolder As IFolder _) As IEnumerable(Of IFileType)
```

**C++**

```cpp
public:
IEnumerable<IFileType>^ GetChildren(_
IFolder^ parentFolder)
```

### Parameters

- **parentFolder**
  - Type: `SimpleEssentials.IO.Types::IFolder`
See Also

FolderHandler Class
SimpleEssentials.IO Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
FolderHandler...Move Method

Namespace: SimpleEssentials.IO
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#
VB
C++

public bool Move(
    ref IFileType file,
    string newPath
)

Public Function Move ( _
    ByRef file As IFileType, _
    newPath As String _
) As Boolean

public:
bool^ Move(
    IFileType^ file,
    String^ newPath
)

Parameters

file
    Type: SimpleEssentials.IO.Types..:.IFileType%

newPath
    Type: String
See Also

FolderHandler Class
SimpleEssentials.IO Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
FolderHandler...Rename Method

Namespace: SimpleEssentials.IO
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

**C#**

public bool Rename(
    ref IFileType file,
    string newName
)

**VB**

Public Function Rename (
    ByRef file As IFileType, _
    newName As String _
) As Boolean

**C++**

public: bool^ Rename(
    IFileType^ file,
    String^ newName
)

**Parameters**

file
   Type: SimpleEssentials.IO.Types:::IFileType%

newName
   Type: String
See Also

FolderHandler Class
SimpleEssentials.IO Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
IFileHandler Interface

Namespace: SimpleEssentials.IO
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#

VB

C++

public interface IFileHandler : IHandler

Public Interface IFileHandler_
    Inherits IHandler

public interface class IFileHandler : IHandler
See Also

IFileHandler Members
SimpleEssentials.IO Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
IFileHandler Members

The IFileHandler type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Inherited From</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Create</strong></td>
<td>Determines whether the specified object is equal to the current object.</td>
<td>Object</td>
</tr>
<tr>
<td><strong>Equals(System.Object)</strong></td>
<td>Determines whether the specified object is equal to the current object.</td>
<td>Object</td>
</tr>
<tr>
<td><strong>Finalize</strong></td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection.</td>
<td>Object</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Serves as the default hash function.</td>
<td>Object</td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td>Gets the <strong>Type</strong> of the current instance.</td>
<td>Object</td>
</tr>
<tr>
<td><strong>MemberwiseClone</strong></td>
<td>Creates a shallow copy of the current <strong>Object</strong>.</td>
<td>Object</td>
</tr>
<tr>
<td><strong>Read(IFile)</strong></td>
<td>Returns a string that represents the current object.</td>
<td></td>
</tr>
<tr>
<td><strong>Read&lt;T&gt;(IFile, IFileReader)</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>ReadAll&lt;T&gt;(IFileReader)</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>ReadBy&lt;T&gt;(IEnumerable&lt;T&gt;, IFileWriter, Boolean)</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Write(IFile, String, Boolean)</strong></td>
<td></td>
<td>Object</td>
</tr>
<tr>
<td><strong>Write&lt;T&gt;(IFile, IFileWriter, Boolean)</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>WriteAll&lt;T&gt;(IEnumerable&lt;T&gt;, IFileWriter, Boolean)</strong></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
See Also

IFileHandler Interface
SimpleEssentials.IO Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
IFileHandler Methods

The IFileHandler type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Create</strong></td>
<td>Determines whether the specified object is equal to the current object. (Inherited from <strong>Object</strong>.)</td>
</tr>
<tr>
<td><strong>Equals(System.Object)</strong></td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from <strong>Object</strong>.)</td>
</tr>
<tr>
<td><strong>Finalize</strong></td>
<td>Serves as the default hash function. (Inherited from <strong>Object</strong>.)</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Gets the <strong>Type</strong> of the current instance. (Inherited from <strong>Object</strong>.)</td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td>Creates a shallow copy of the current <strong>Object</strong>. (Inherited from <strong>Object</strong>.)</td>
</tr>
<tr>
<td><strong>MemberwiseClone</strong></td>
<td>Returns a string that represents the current object. (Inherited from <strong>Object</strong>.)</td>
</tr>
<tr>
<td><strong>Read(IFile)</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Read&lt;(Of &lt;&lt;'(T)&gt;&gt;)(IFile, IFileReader)</strong></td>
<td></td>
</tr>
<tr>
<td><strong>ReadAll&lt;(Of &lt;&lt;'(T)&gt;&gt;)</strong></td>
<td></td>
</tr>
<tr>
<td><strong>ReadBy&lt;(Of &lt;&lt;'(T)&gt;&gt;)</strong></td>
<td></td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Write(IFile, String, Boolean)</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Write&lt;(Of &lt;&lt;'(T)&gt;&gt;)(IFile, T, IFileWriter, Boolean)</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Write&lt;(Of &lt;&lt;'(T)&gt;&gt;)(IFile, IEnumerable&lt;(Of &lt;&lt;'(T)&gt;&gt;), IFileWriter, Boolean)</strong></td>
<td></td>
</tr>
</tbody>
</table>
See Also

IFileHandler Interface
SimpleEssentials.IO Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
IFileHandler Create Method

Namespace:  SimpleEssentials.IO
Assembly:  SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#

VB

C++

IFile Create(
    string fileName,
    IFolder parentFolder
)

Function Create ( _
    fileName As String, _
    parentFolder As IFolder _
) As IFile

IFile^ Create(
    String^ fileName,
    IFolder^ parentFolder
)

Parameters

fileName
    Type: String

parentFolder
    Type: SimpleEssentials.IO.Types...IFolder
See Also

IFileHandler Interface
SimpleEssentials.IO Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
IFileHandler...::..Read Method (IFile)

Namespace: SimpleEssentials.IO
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#
VB
C++

```csharp
string Read(IFile file)
```

```vbnet
Function Read(file As IFile) As String
```

```cpp
String^ Read(IFile^ file)
```

Parameters

file
    Type: SimpleEssentials.IO.Types..:..IFile
See Also

IFileHandler Interface
Read Overload
SimpleEssentials.IO Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
IFileHandler...:::Read<(Of (<'T'>))> Method (IFile, IFileReader)

Namespace: SimpleEssentials.IO
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#

VB

C++

T Read<T>(
    IFile file,
    IFileReader fileReader
)

Function Read(Of T) ( _
    file As IFile, _
    fileReader As IFileReader _
) As T

generic<typename T>

T Read(
    IFile^ file,
    IFileReader^ fileReader
)

Type Parameters

T

Parameters

file
    Type: SimpleEssentials.IO.Types..::.IFile

fileReader
    Type: SimpleEssentials.IO.Readers..::.IFileReader
See Also

IFileHandler Interface
Read Overload
SimpleEssentials.IO Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
IFileHandler...::.ReadAll(Of "T")> Method

Namespace: SimpleEssentials.IO
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#  
VB  
C++

```csharp
IEnumerable<T> ReadAll<T>(
    IFile file,
    IFileReader fileReader
)
```

```vbnet
Function ReadAll(Of T) ( _
    file As IFile, _
    fileReader As IFileReader _
) As IEnumerable(Of T)
```

generic<tpm T>

```csharp
IEnumerable<T>^ ReadAll(
    IFile^ file,
    IFileReader^ fileReader
)
```

Type Parameters

T

Parameters

file
    Type: SimpleEssentials.IO.Types..::..IFile

fileReader
    Type: SimpleEssentials.IO.Readers..::..IFileReader
See Also

IFileHandler Interface  
SimpleEssentials.IO Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
IFileHandler...::..ReadBy<(Of <('T')>)> Method

Namespace: SimpleEssentials.IO
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#

```csharp
IEnumerable<T> ReadBy<T>(
    IFile file,
    Func<T, bool> predicate,
    IFileReader fileReader
)
```

Function `ReadBy(Of T)` ( _
    file As IFile, _
    predicate As Func(Of T, Boolean), _
    fileReader As IFileReader _
) As IEnumerable(Of T)

generic<typename T>
```csharp
IEnumerable<T>^ ReadBy(
    IFile^ file,
    Func<T, bool>^ predicate,
    IFileReader^ fileReader
)
```

Type Parameters

T

Parameters

file
    Type: SimpleEssentials.IO.Types..::.IFile

predicate
    Type: Func(Of ((<T, Boolean>))>

fileReader
    Type: SimpleEssentials.IO.Readers..::.IFileReader
See Also

IFileHandler Interface
SimpleEssentials.IO Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
IFileHandler...:::Write Method
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Write(IFile, String, Boolean)</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Write(Of &lt;&lt;(T)&gt;&gt;)(IFile, T, IFileWriter, Boolean)</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Write(Of &lt;&lt;(T)&gt;&gt;)(IFile, IEnumerable(Of &lt;&lt;(T)&gt;&gt;), IFileWriter, Boolean)</strong></td>
<td></td>
</tr>
</tbody>
</table>
See Also

IFileHandler Interface
IFileHandler Members
SimpleEssentials.IO Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
IFileHandler...:::Write Method (IFile, String, Boolean)

Namespace: SimpleEssentials.IO
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#
VB
C++

```csharp
bool Write(
    IFile file, 
    string content, 
    bool append
) )
```

```vb
Function Write (_, 
    file As IFile, _, 
    content As String, _ 
    append As Boolean _) 
) As Boolean
```

```cpp
bool^ Write( 
    IFile^ file, 
    String^ content, 
    bool^ append 
) 
```

Parameters

file
    Type: SimpleEssentials.IO.Types..:::IFile

ccontent
    Type: String

append
    Type: Boolean
See Also

IFileHandler Interface
Write Overload
SimpleEssentials.IO Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
IFileHandler...:::Write(Of <('T')>)> Method (IFile, T, IFileWriter, Boolean)

Namespace: SimpleEssentials.IO
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#
VB
C++

void Write<T>(
    IFile file,
    T obj,
    IFileWriter fileWriter,
    bool append
)

Sub Write(Of T) ( _
    file As IFile, _
    obj As T, _
    fileWriter As IFileWriter, _
    append As Boolean _
)

generic<typename T>
void Write(
    IFile^ file,
    T obj,
    IFileWriter^ fileWriter,
    bool^ append
)

Type Parameters

T

Parameters

file
    Type: SimpleEssentials.IO.Types.IFile

obj
    Type: T
fileWriter
   Type: SimpleEssentials.IO.Writers.IO.FileWriter

append
   Type: Boolean
See Also

IFileHandler Interface
Write Overload
SimpleEssentials.IO Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
IFileHandler...:::Write<(Of <('T')>)> Method (IFile, IEnumerable<(Of <('T')>)), IFileWriter, Boolean)

Namespace:  SimpleEssentials.IO
Assembly:  SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#

```csharp
void Write<T>(
    IFile file,
    IEnumerable<T> obj,
    IFileWriter fileWriter,
    bool append
)
```

VB

```vbnet
Sub Write(Of T) ( _
    file As IFile, _
    obj As IEnumerable(Of T), _
    fileWriter As IFileWriter, _
    append As Boolean _
)
```

generic<typename T>

```csharp
void Write(
    IFile^ file,
    IEnumerable<T>^ obj,
    IFileWriter^ fileWriter,
    bool^ append
)
```

Type Parameters

T

Parameters

file
Type: `SimpleEssentials.IO.Types.IFile`

obj
Type: `IEnumerable<T>`
fileWriter
    Type: SimpleEssentials.IO.Writers....IFileWriter

append
    Type: Boolean
See Also

IFileHandler Interface
Write Overload
SimpleEssentials.IO Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
IFolderHandler Interface

Namespace: SimpleEssentials.IO
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

**C#**

```csharp
public interface IFolderHandler : IHandler

Public Interface IFolderHandler _
    Inherits IHandler

public interface class IFolderHandler : IHandler
```

**VB**

```vbnet
Public Interface IFolderHandler Inherits IHandler

Public Interface IFolderHandler : IHandler
```

**C++**

```cpp
public interface IFolderHandler : IHandler

public interface class IFolderHandler : IHandler
```
See Also

IFolderHandler Members
SimpleEssentials.IO Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
IFolderHandler Members

The IFolderHandler type exposes the following members.
### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Create(String, Boolean)</code></td>
<td>Determines whether the specified object is equal to the current object.</td>
</tr>
<tr>
<td><code>Create(String, IFolder)</code></td>
<td>(Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><code>Equals(System.Object)</code></td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection.</td>
</tr>
<tr>
<td><code>Finalize</code></td>
<td>(Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><code>GetChildFiles</code></td>
<td>Serves as the default hash function.</td>
</tr>
<tr>
<td><code>GetChildFolders</code></td>
<td>(Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><code>GetChildren</code></td>
<td>Gets the <code>Type</code> of the current instance.</td>
</tr>
<tr>
<td><code>GetHashCode</code></td>
<td>(Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><code>GetType</code></td>
<td>Creates a shallow copy of the current <code>Object</code>.</td>
</tr>
<tr>
<td><code>MemberwiseClone</code></td>
<td>(Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><code>ToString</code></td>
<td>Returns a string that represents the current object.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <code>Object</code>.)</td>
</tr>
</tbody>
</table>
See Also

IFolderHandler Interface
SimpleEssentials.IO Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
IFolderHandler Methods

The IFolderHandler type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create(String, Boolean)</td>
<td>Determines whether the specified object is equal to the current object.</td>
</tr>
<tr>
<td>Create(String, IFolder)</td>
<td>(Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td>Equals(System.Object)</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection.</td>
</tr>
<tr>
<td>finalize</td>
<td>(Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td>GetChildFiles</td>
<td>Serves as the default hash function.</td>
</tr>
<tr>
<td>GetChildFolders</td>
<td>(Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td>GetChildren</td>
<td>Gets the <a href="#">Type</a> of the current instance.</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>(Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Creates a shallow copy of the current <a href="#">Object</a>.</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>(Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">Object</a>.)</td>
</tr>
</tbody>
</table>
See Also

IFolderHandler Interface
SimpleEssentials.IO Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
IFolderHandler..::..Create Method
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create(String, Boolean)</td>
<td></td>
</tr>
<tr>
<td>Create(String, IFolder)</td>
<td></td>
</tr>
</tbody>
</table>
See Also

IFolderHandler Interface
IFolderHandler Members
SimpleEssentials.IO Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
IFolderHandler::Create Method (String, Boolean)

Namespace: SimpleEssentials.IO
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#
VB
C++

```csharp
IFolder Create(
    string path,
    bool relative
)
```

```vb
Function Create ( _
    path As String, _
    relative As Boolean _
) As IFolder
```

```cpp
IFolder^ Create(
    String^ path,
    bool^ relative
)
```

Parameters

path
Type: String

relative
Type: Boolean
See Also

IFolderHandler Interface
Create Overload
SimpleEssentials.IO Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
IFolderHandler...Create Method (String, IFolder)

Namespace: SimpleEssentials.IO
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#

VB

C++

IFolder Create(
    string path,
    IFolder parent
)

Function Create ( _
    path As String, _
    parent As IFolder _
) As IFolder

IFolder^ Create(
    String^ path,
    IFolder^ parent
)

Parameters

path
    Type: String

parent
    Type: SimpleEssentials.IO.Types:::IFolder
See Also

IFolderHandler Interface
Create Overload
SimpleEssentials.IO Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
IFolderHandler..::..GetChildFiles Method

**Namespace:**  SimpleEssentials.IO
**Assembly:**  SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#

```csharp
IEnumerable<IFile> GetChildFiles(IFolder parentFolder)
```

VB

```vbnet
Function GetChildFiles(parentFolder As IFolder) As IEnumerable(Of IFile)
```

C++

```cpp
IEnumerable<IFile> GetChildFiles(IFolder parentFolder)
```

Parameters

parentFolder
  Type: SimpleEssentials.IO.Types::IFolder
See Also

IFolderHandler Interface
SimpleEssentials.IO Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
IFolderHandler..::..GetChildFolders Method

Namespace: SimpleEssentials.IO
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#
VB
C++

IEnumerable<IFolder> GetChildFolders(
    IFolder parentFolder
)

Function GetChildFolders ( _
    parentFolder As IFolder _
) As IEnumerable(Of IFolder)

IEnumerable<IFolder> GetChildFolders(
    IFolder parentFolder
)

Parameters

parentFolder
    Type: SimpleEssentials.IO.Types:::IFolder
See Also

IFolderHandler Interface
SimpleEssentials.IO Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
IFolderHandler...GetChildren

**Method**

**Namespace:** SimpleEssentials.IO  
**Assembly:** SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#
VB
C++

IEnumerable<IFileType> GetChildren(
    IFolder parentFolder
)

Function GetChildren ( _
    parentFolder As IFolder _
) As IEnumerable(Of IFileType)

IEnumerable<IFileType>^ GetChildren(
    IFolder^ parentFolder
)

Parameters

parentFolder
    Type: SimpleEssentials.IO.Types...IFolder
See Also

IFolderHandler Interface
SimpleEssentials.IO Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
IHandler Interface

Namespace:  SimpleEssentials.IO
Assembly:  SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#
VB
C++

public interface IHandler
Public Interface IHandler
public interface class IHandler
See Also

IHandler Members
SimpleEssentials.IO Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
IHandler Members

The IHandler type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>✿ Create</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>✿ Equals(System.Object)</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)</td>
</tr>
<tr>
<td>✿ Finalize</td>
<td></td>
</tr>
<tr>
<td>✿ Get</td>
<td></td>
</tr>
<tr>
<td>✿ GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>✿ GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>✿ MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>✿ Move</td>
<td></td>
</tr>
<tr>
<td>✿ Rename</td>
<td></td>
</tr>
<tr>
<td>✿ ToString</td>
<td>Returns a string that represents the current object. (Inherited from Object.)</td>
</tr>
</tbody>
</table>
See Also

IHandler Interface
SimpleEssentials.IO Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
IHHandler Methods

The IHHandler type exposes the following members.
# Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>📔 Create</td>
<td>Determines whether the specified object is equal to the current object.</td>
</tr>
<tr>
<td>🤐 Equals(System.Object)</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection.</td>
</tr>
<tr>
<td>🚧 Finalize</td>
<td>Serves as the default hash function.</td>
</tr>
<tr>
<td>🍀 Get</td>
<td>Gets the Type of the current instance.</td>
</tr>
<tr>
<td>🍀 GetHashCode</td>
<td>Creates a shallow copy of the current Object.</td>
</tr>
<tr>
<td>🍀 GetType</td>
<td>Returns a string that represents the current object.</td>
</tr>
<tr>
<td>🍀 Move</td>
<td>(Inherited from Object.)</td>
</tr>
<tr>
<td>🍀 Rename</td>
<td>(Inherited from Object.)</td>
</tr>
<tr>
<td>🍀 ToString</td>
<td>(Inherited from Object.)</td>
</tr>
</tbody>
</table>
See Also

IHandler Interface
SimpleEssentials.IO Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
IHandler..::..Create Method

Namespace:  SimpleEssentials.IO
Assembly:  SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#
VB
C++

`IFileType` Create(
    `string` path
)

Function Create ( _
    path As `String` _
) As `IFileType`

`IFileType` Create(
    `String`^ path
)

**Parameters**

path
    Type: `String`
See Also

IHandler Interface
SimpleEssentials.IO Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
IHandler...::..GetMethod

Namespace: SimpleEssentials.IO
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#

)`

VB

C++

IFileType Get(
    String path
)

Function Get ( _
    path As String _
) As IFileType

IFileType^ Get(
    String^ path
)

Parameters

path
    Type: String
See Also

IHandler Interface
SimpleEssentials.IO Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
IHandler..::..Move Method

Namespace:  SimpleEssentials.IO
Assembly:  SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#

```csharp
bool Move(
    ref IFileType file,
    string newPath
)
```

VB

```vb
Function Move (_
    ByRef file As IFileType, _
    newPath As String _
) As Boolean
```

C++

```cpp
bool Move(
    IFileType% file,
    String% newPath
)
```

Parameters

file
Type: SimpleEssentials.IO.Types::IFileType%

newPath
Type: String
See Also

IHامل Interface
SimpleEssentials.IO Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
IHandler..::..Rename Method

Namespace: SimpleEssentials.IO
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#

```csharp
bool Rename(
    ref IFileType file,
    string newName
)
```

Function Rename ( _
    ByRef file As IFileType, _
    newName As String _
) As Boolean

```vbnet
bool^ Rename(
    IFileType^% file,
    String^% newName
)
```

Parameters

file
    Type: SimpleEssentials.IO.Types..::..IFileType%

newName
    Type: String
See Also

IHandler Interface
SimpleEssentials.IO Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
SimpleEssentials.IO.Readers
Namespace
# Classes

<table>
<thead>
<tr>
<th>Class</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>🌟 CsvReader</td>
<td></td>
</tr>
</tbody>
</table>
## Interfaces

<table>
<thead>
<tr>
<th>Interface</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IFileReader</td>
<td></td>
</tr>
</tbody>
</table>

*Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.*
CsvReader Class

Namespace:  SimpleEssentials.IO.Readers
Assembly:  SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#  
Public class CsvReader : IFileReader

VB  
Public Class CsvReader  
    Implements IFileReader

C++  
public ref class CsvReader : IFileReader
Inheritance Hierarchy

Object
SimpleEssentials.IO.Readers..CsvReader
See Also

CsvReader Members
SimpleEssentials.IO.Readers Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
CsvReader Members

The CsvReader type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Equals(System.Object)</strong></td>
<td>Determines whether the specified object is equal to the current object. (Inherited from <strong>Object</strong>.)</td>
</tr>
<tr>
<td><strong>Finalize</strong></td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from <strong>Object</strong>.)</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Serves as the default hash function. (Inherited from <strong>Object</strong>.)</td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td>Gets the <strong>Type</strong> of the current instance. (Inherited from <strong>Object</strong>.)</td>
</tr>
<tr>
<td><strong>MemberwiseClone</strong></td>
<td>Creates a shallow copy of the current <strong>Object</strong>. (Inherited from <strong>Object</strong>.)</td>
</tr>
<tr>
<td><strong>Read(Of &lt;&lt;(T)&gt;&gt;)</strong></td>
<td></td>
</tr>
<tr>
<td><strong>ReadAll(Of &lt;&lt;(T)&gt;&gt;)</strong></td>
<td></td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td>Returns a string that represents the current object. (Inherited from <strong>Object</strong>.)</td>
</tr>
</tbody>
</table>
See Also

CsvReader Class
SimpleEssentials.IO.Readers Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
CsvReader Methods

The CsvReader type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Equals(System.Object)</strong></td>
<td>Determines whether the specified object is equal to the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td><strong>Finalize</strong></td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td><strong>MemberwiseClone</strong></td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td><strong>Read&lt;T&gt;</strong></td>
<td>Returns a string that represents the current object. (Inherited from Object.)</td>
</tr>
</tbody>
</table>
See Also

CsvReader Class
SimpleEssentials.IO.Readers Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
CsvReader..::.Read(Of ("T"))

Method

Namespace: SimpleEssentials.IO.Readers
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#
VB
C++

public T Read<T>(
        string filePath
    )

Public Function Read(Of T) ( _
            filePath As String _
    ) As T

public:
generic<typename T>
T Read(
        String^ filePath
    )

Type Parameters

T

Parameters

filePath
    Type: String
See Also

CsvReader Class
SimpleEssentials.IO.Readers Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
CsvReader..::..ReadAll<(Of <('T')>)> Method

Namespace: SimpleEssentials.IO.Readers
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#

```csharp
public IEnumerable<T> ReadAll<T>(
    string filePath
)
```

VB

```vbnet
Public Function ReadAll(Of T) ( _
    filePath As String _
) As IEnumerable(Of T)
```

C++

```cpp
public:
    template<typename T>
    IEnumerable<T>^ ReadAll( _
        String^ filePath
    )
```

Type Parameters

T

Parameters

filePath
    Type: String
See Also

CsvReader Class
SimpleEssentials.IO.Readers Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
IFileReader Interface

Namespace: SimpleEssentials.IO.Readers
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#
VB
C++

public interface IFileReader

Public Interface IFileReader

public interface class IFileReader
See Also

IFileReader Members
SimpleEssentials.IO.Readers Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
IFileReader Members

The IFileReader type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Equals(System.Object)</strong></td>
<td>Determines whether the specified object is equal to the current object.</td>
</tr>
<tr>
<td>(Inherited from <strong>Object</strong>.)</td>
<td></td>
</tr>
<tr>
<td><strong>Finalize</strong></td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection.</td>
</tr>
<tr>
<td>(Inherited from <strong>Object</strong>.)</td>
<td></td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Serves as the default hash function.</td>
</tr>
<tr>
<td>(Inherited from <strong>Object</strong>.)</td>
<td></td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td>Gets the <strong>Type</strong> of the current instance.</td>
</tr>
<tr>
<td>(Inherited from <strong>Object</strong>.)</td>
<td></td>
</tr>
<tr>
<td><strong>MemberwiseClone</strong></td>
<td>Creates a shallow copy of the current <strong>Object</strong>.</td>
</tr>
<tr>
<td>(Inherited from <strong>Object</strong>.)</td>
<td></td>
</tr>
<tr>
<td><strong>Read&lt;T&gt;()</strong></td>
<td>Returns a string that represents the current object.</td>
</tr>
<tr>
<td>(Inherited from <strong>Object</strong>.)</td>
<td></td>
</tr>
<tr>
<td><strong>ReadAll&lt;T&gt;()</strong></td>
<td></td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td></td>
</tr>
</tbody>
</table>
See Also

IFileReader Interface
SimpleEssentials.IO.Readers Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
IFileReader Methods

The IFileReader type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals(System.Object)</td>
<td>Determines whether the specified object is equal to the current object.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from Object.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup</td>
</tr>
<tr>
<td></td>
<td>operations before it is reclaimed by garbage collection.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from Object.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from Object.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from Object.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from Object.)</td>
</tr>
<tr>
<td>Read&lt;(Of &lt;&lt;'(T)&gt;&gt;)&gt;</td>
<td>Returns a string that represents the current object.</td>
</tr>
<tr>
<td>ReadAll&lt;(Of &lt;&lt;'(T)&gt;&gt;)&gt;</td>
<td>(Inherited from Object.)</td>
</tr>
<tr>
<td>ToString</td>
<td></td>
</tr>
</tbody>
</table>
See Also

IFileReader Interface
SimpleEssentials.IO.Readers Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
IFileReader..::.Read(Of <('T)>)> Method

Namespace: SimpleEssentials.IO.Readers  
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

**C#**

```csharp
T Read<T>(
    string filePath
)
```

**VB**

```vbnet
Function Read(Of T) ( _
    filePath As String _
) As T

generic<Typename T>

T Read(
    String^ filePath
)
```

**C++**

```cpp
generic<typename T>
T Read(
    String^ filePath
)
```

**Type Parameters**

T

**Parameters**

filePath
    Type: String
See Also

IFileReader Interface
SimpleEssentials.IO.Reade\_\_ers Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
**IFileReader..::.ReadAll(Of (<'T'>)>) Method**

**Namespace:** SimpleEssentials.IO.Readers

**Assembly:** SimpleEssentials (in SimpleEssentials.dll)
Syntax

**C#**

```csharp
IEnumerable<T> ReadAll<T>(
    string filePath
)
```

**VB**

```vbnet
Function ReadAll(Of T) ( _
    filePath As String _
) As IEnumerable(Of T)
```

**C++**

```cpp
generic<typename T>
IEnumerable<T>^ ReadAll(
    String^ filePath
)
```

**Type Parameters**

T

**Parameters**

**filePath**
    Type: String
See Also

IFileReader Interface
SimpleEssentials.IO.Readers Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
SimpleEssentials.IO.Types
Namespace
# Classes

<table>
<thead>
<tr>
<th>Class Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>File</td>
</tr>
<tr>
<td>Folder</td>
</tr>
</tbody>
</table>
Interfaces

Interface Description

- IFile
- IFileType
- IFolder
## Enumerations

<table>
<thead>
<tr>
<th>Enumeration Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>FileType</strong></td>
</tr>
</tbody>
</table>

*Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.*
File Class

Namespace:  SimpleEssentials.IO.Types
Assembly:  SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#

VB

C++

public class File : IFile

Public Class File _
    Implements IFile

public ref class File : IFile
Inheritance Hierarchy

Object
SimpleEssentials.IO.Types......File
See Also

File Members
SimpleEssentials.IO.Types Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
File Members

The File type exposes the following members.
## Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>File()</td>
<td></td>
</tr>
<tr>
<td>File(String)</td>
<td></td>
</tr>
</tbody>
</table>
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Equals(System.Object)</code></td>
<td>Determines whether the specified object is equal to the current object. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><code>Exist</code></td>
<td></td>
</tr>
<tr>
<td><code>Finalize</code></td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><code>GetHashCode</code></td>
<td>Serves as the default hash function. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><code>GetType</code></td>
<td>Gets the <code>Type</code> of the current instance. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><code>Load</code></td>
<td></td>
</tr>
<tr>
<td><code>MemberwiseClone</code></td>
<td>Creates a shallow copy of the current <code>Object</code>. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><code>ToString</code></td>
<td>Returns a string that represents the current object. (Inherited from <code>Object</code>.)</td>
</tr>
</tbody>
</table>
# Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Extension</td>
<td></td>
</tr>
<tr>
<td>FullPath</td>
<td></td>
</tr>
<tr>
<td>Loaded</td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td></td>
</tr>
<tr>
<td>Size</td>
<td></td>
</tr>
<tr>
<td>Type</td>
<td></td>
</tr>
</tbody>
</table>
See Also

File Class
SimpleEssentials.IO.Types Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
File Constructor
Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>File()</td>
<td></td>
</tr>
<tr>
<td>File(String)</td>
<td></td>
</tr>
</tbody>
</table>
See Also

File Class
File Members
SimpleEssentials.IO.Types Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
File Constructor

Namespace:  SimpleEssentials.IO.Types
Assembly:  SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#  
VB  
C++

public File()

Public Sub New

public:
File()
See Also

File Class
File Overload
SimpleEssentials.IO.Types Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
File Constructor (String)

Namespace: SimpleEssentials.IO.Types
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#  
public File(
    string path
)

VB  
Public Sub New ( _
    path As String _
)

C++  
public:  
File(
    String^ path
)

Parameters

path  
Type: String
See Also

File Class
File Overload
SimpleEssentials.IO.Types Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
File Methods

The File type exposes the following members.
# Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Equals(System.Object)</strong></td>
<td>Determines whether the specified object is equal to the current object. (Inherited from <strong>Object</strong>.)</td>
</tr>
<tr>
<td><strong>Exist</strong></td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from <strong>Object</strong>.)</td>
</tr>
<tr>
<td><strong>Finalize</strong></td>
<td>Serves as the default hash function. (Inherited from <strong>Object</strong>.)</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Gets the <strong>Type</strong> of the current instance. (Inherited from <strong>Object</strong>.)</td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td>Creates a shallow copy of the current <strong>Object</strong>. (Inherited from <strong>Object</strong>.)</td>
</tr>
<tr>
<td><strong>Load</strong></td>
<td>Returns a string that represents the current object. (Inherited from <strong>Object</strong>.)</td>
</tr>
</tbody>
</table>
See Also

File Class
SimpleEssentials.IO.Types Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
File..::..Exist Method

Namespace: SimpleEssentials.IO.Types
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#

```csharp
public bool Exist(string path)
```

VB

```vbnet
Public Function Exist(_
    path As String _
) As Boolean
```

C++

```cpp
public: bool Exist(_
    String^ path
)
```

Parameters

path

Type: String
See Also

File Class
SimpleEssentials.IO.Types Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
File...Load Method

Namespace: SimpleEssentials.IO.Types
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#
VB
C++

public bool Load(
    string path
)

Public Function Load (_
    path As String _
) As Boolean

public:
bool^ Load(
    String^ path
)

Parameters

path
    Type: String
See Also

**File Class**
**SimpleEssentials.IO.Types Namespace**

*Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.*
File Properties

The `File` type exposes the following members.
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Extension</td>
<td></td>
</tr>
<tr>
<td>FullPath</td>
<td></td>
</tr>
<tr>
<td>Loaded</td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td></td>
</tr>
<tr>
<td>Size</td>
<td></td>
</tr>
<tr>
<td>Type</td>
<td></td>
</tr>
</tbody>
</table>
See Also

File Class
SimpleEssentials.IO.Types Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
File...Extension Property

Namespace: SimpleEssentials.IO.Types
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#  
VB  
C++  

public string Extension { get; set; }

Public Property Extension As String
  Get
  Set

public:
property String^ Extension {
  String^ get ();
  void set (String^ value);
}
See Also

File Class
SimpleEssentials.IO.Types Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
File:::.FullPath Property

Namespace: SimpleEssentials.IO.Types
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

**C#**

```csharp
public string FullPath { get; set; }
```

**VB**

```vbnet
Public Property FullPath As String
    Get
    Set
```

**C++**

```cpp
public:
    property String^ FullPath {
        String^ get ();
        void set (String^ value);
    }
```
See Also

File Class
SimpleEssentials.IO.Types Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
File...:::Loaded Property

Namespace: SimpleEssentials.IO.Types
Assembly: SimpleEssentials (in SimpleEssentials.dll)
**Syntax**

C#

```csharp
public bool Loaded { get; set; }
```

VB

```vbnet
Public Property Loaded As Boolean
    Get
        Set
    End Property
```

C++

```c++
public: bool^ Loaded {
    bool^ get ();
    void set (bool^ value);
}
```
See Also

File Class
SimpleEssentials.IO.Types Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
File:::Name Property

**Namespace:** SimpleEssentials.IO.Types  
**Assembly:** SimpleEssentials (in SimpleEssentials.dll)
Syntax

**C#**

```csharp
public string Name { get; set; }
```

**VB**

```vbnet
Public Property Name As String
    Get
    Set
```

**C++**

```cpp
public:
    property String^ Name {
        String^ get ();
        void set (String^ value);
    }
```
See Also

File Class
SimpleEssentials.IO.Types Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
File Size Property

Namespace: SimpleEssentials.IO.Types
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#

VB

C++

public long Size { get; set; }

Public Property Size As Long
    Get
    Set

public:
property long long^ Size {
    long long^ get ();
    void set (long long^ value);
}
See Also

File Class
SimpleEssentials.IO.Types Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
File..::.Type Property

Namespace:  SimpleEssentials.IO.Types
Assembly:  SimpleEssentials (in SimpleEssentials.dll)
Syntax

public FileType Type { get; }

Public ReadOnly Property Type As FileType
    Get

public:
    property FileType^ Type {
        FileType^ get ();
    }

See Also

File Class
SimpleEssentials.IO.Types Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
FileType Enumeration

Namespace: SimpleEssentials.IO.Types
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#
VB
C++

public enum FileType
Public Enumeration FileType
public enum class FileType
## Members

<table>
<thead>
<tr>
<th>Member name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FILE</td>
<td></td>
</tr>
<tr>
<td>FOLDER</td>
<td></td>
</tr>
</tbody>
</table>
See Also

SimpleEssentials.IO.Types Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
Folder Class

Namespace:  SimpleEssentials.IO.Types
Assembly:  SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#

public class Folder : IFolder

VB

Public Class Folder _
    Implements IFolder

C++

public ref class Folder : IFolder
Inheritance Hierarchy

Object
SimpleEssentials.IO.Types...Folder
See Also

Folder Members
SimpleEssentials.IO.Types Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
Folder Members

The Folder type exposes the following members.
Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Folder()()()()</td>
<td></td>
</tr>
<tr>
<td>Folder(String)</td>
<td></td>
</tr>
</tbody>
</table>
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Equals(System.Object)</strong></td>
<td>Determines whether the specified object is equal to the current object. (Inherited from <strong>Object</strong>.)</td>
</tr>
<tr>
<td><strong>Exist</strong></td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from <strong>Object</strong>.)</td>
</tr>
<tr>
<td><strong>Finalize</strong></td>
<td>Serves as the default hash function. (Inherited from <strong>Object</strong>.)</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Gets the <strong>Type</strong> of the current instance. (Inherited from <strong>Object</strong>.)</td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td>Creates a shallow copy of the current <strong>Object</strong>. (Inherited from <strong>Object</strong>.)</td>
</tr>
<tr>
<td><strong>Load</strong></td>
<td>Returns a string that represents the current object. (Inherited from <strong>Object</strong>.)</td>
</tr>
</tbody>
</table>
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FullPath</td>
<td></td>
</tr>
<tr>
<td>Loaded</td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td></td>
</tr>
<tr>
<td>Type</td>
<td></td>
</tr>
</tbody>
</table>
See Also

Folder Class
SimpleEssentials.IO.Types Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
Folder Constructor
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Folder()</code></td>
<td></td>
</tr>
<tr>
<td><code>Folder(String)</code></td>
<td></td>
</tr>
</tbody>
</table>
See Also

Folder Class
Folder Members
SimpleEssentials.IO.Types Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
Folder Constructor

Namespace: SimpleEssentials.IO.Types
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

### C#
```
public Folder()
```

### VB
```
Public Sub New
```

### C++
```
public:
Folder()
```
See Also

Folder Class
Folder Overload
SimpleEssentials.IO.Types Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
Folder Constructor (String)

Namespace: SimpleEssentials.IO.Types
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#
VB
C++

public Folder(
    string path
)

Public Sub New ( _
    path As String _
)

public:
Folder(
    String^ path
)

Parameters

path
    Type: String
See Also

Folder Class
Folder Overload
SimpleEssentials.IO.Types Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
Folder Methods

The Folder type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>✡ <strong>Equals(System.Object)</strong></td>
<td>Determines whether the specified object is equal to the current object. (Inherited from <strong>Object</strong>.)</td>
</tr>
<tr>
<td>✡ <strong>Exist</strong></td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from <strong>Object</strong>.)</td>
</tr>
<tr>
<td>✡ <strong>Finalize</strong></td>
<td>Serves as the default hash function. (Inherited from <strong>Object</strong>.)</td>
</tr>
<tr>
<td>✡ <strong>GetHashCode</strong></td>
<td>Gets the <strong>Type</strong> of the current instance. (Inherited from <strong>Object</strong>.)</td>
</tr>
<tr>
<td>✡ <strong>GetType</strong></td>
<td>Creates a shallow copy of the current <strong>Object</strong>. (Inherited from <strong>Object</strong>.)</td>
</tr>
<tr>
<td>✡ <strong>Load</strong></td>
<td>Returns a string that represents the current object. (Inherited from <strong>Object</strong>.)</td>
</tr>
<tr>
<td>✡ <strong>MemberwiseClone</strong></td>
<td></td>
</tr>
<tr>
<td>✡ <strong>ToString</strong></td>
<td></td>
</tr>
</tbody>
</table>
See Also

Folder Class
SimpleEssentials.IO.Types Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
Folder...Exist Method

Namespace: SimpleEssentials.IO.Types
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#  
public bool Exist(
    string path
)

VB  
Public Function Exist ( _
    path As String _
) As Boolean

c++  
public:
bool^ Exist(
    String^ path
)

Parameters

path  
Type: String
See Also

Folder Class
SimpleEssentials.IO.Types Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
Folder...:..Load Method

Namespace: SimpleEssentials.IO.Types
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#
VB
C++

public bool Load(
    string path
)

Public Function Load ( _
    path As String _
) As Boolean

public:
bool^ Load(
    String^ path
)

Parameters

path
    Type: String
See Also

Folder Class
SimpleEssentials.IO.Types Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
Folder Properties

The **Folder** type exposes the following members.
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FullPath</td>
<td></td>
</tr>
<tr>
<td>Loaded</td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td></td>
</tr>
<tr>
<td>Type</td>
<td></td>
</tr>
</tbody>
</table>
See Also

Folder Class
SimpleEssentials.IO.Types Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
Folder..::..FullPath Property

Namespace: SimpleEssentials.IO.Types
Assembly: SimpleEssentials (in SimpleEssentials.dll)
**Syntax**

**C#**
```
public string FullPath { get; set; }
```

**VB**
```
Public Property FullPath As String
    Get
    Set
```

**C++**
```
public:
    property String^ FullPath {
        String^ get ();
        void set (String^ value);
    }
```
See Also

Folder Class
SimpleEssentials.IO.Types Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
Folder..::..Loaded Property

**Namespace:**  [SimpleEssentials.IO.Types]

**Assembly:**  SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#

public bool Loaded { get; set; }

VB

Public Property Loaded As Boolean
    Get
    Set

C++

public:
    property bool^ Loaded {
        bool^ get();
        void set (bool^ value);
    }
See Also

Folder Class
SimpleEssentials.IO.Types Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
Folder...Name Property

**Namespace:** SimpleEssentials.IO.Types  
**Assembly:** SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#

```csharp
public string Name { get; set; }
```

VB

```vbnet
Public Property Name As String
    Get
    Set
```

C++

```cpp
public:
    property String^ Name {
        String^ get ();
        void set (String^ value);
    }
```
See Also

Folder Class
SimpleEssentials.IO.Types Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
Folder..::..Type Property

Namespace: SimpleEssentials.IO.Types
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#
VB
C++

public FileType Type { get; }

Public ReadOnly Property Type As FileType
    Get

public:
    property FileType^ Type {
        FileType^ get ();
    }

}
See Also

Folder Class
SimpleEssentials.IO.Types Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
IFile Interface

Namespace:  SimpleEssentials.IO.Types
Assembly:  SimpleEssentials (in SimpleEssentials.dll)
## Syntax

### C#
```csharp
public interface IFile : IFileType
```

### VB
```vbnet
Public Interface IF ile
    Inherits IFileType
```

### C++
```c++
public interface class IFile : IFileType
```

See Also

IFile Members
SimpleEssentials.IO.Types Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
IFile Members

The IFile type exposes the following members.
# Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Equals(System.Object)</code></td>
<td>Determines whether the specified object is equal to the current object. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><code>Finalize</code></td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><code>GetHashCode</code></td>
<td>Serves as the default hash function. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><code>GetType</code></td>
<td>Gets the <code>Type</code> of the current instance. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><code>MemberwiseClone</code></td>
<td>Creates a shallow copy of the current <code>Object</code>. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><code>ToString</code></td>
<td>Returns a string that represents the current object. (Inherited from <code>Object</code>.)</td>
</tr>
</tbody>
</table>
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Extension</td>
<td></td>
</tr>
<tr>
<td>Size</td>
<td></td>
</tr>
</tbody>
</table>
See Also

IFile Interface
SimpleEssentials.IO.Types Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
IFile Methods

The IFile type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Equals(System.Object)</code></td>
<td>Determines whether the specified object is equal to the current object. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><code>Finalize</code></td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><code>GetHashCode</code></td>
<td>Serves as the default hash function. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><code>GetType</code></td>
<td>Gets the <code>Type</code> of the current instance. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><code>MemberwiseClone</code></td>
<td>Creates a shallow copy of the current <code>Object</code>. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><code>ToString</code></td>
<td>Returns a string that represents the current object. (Inherited from <code>Object</code>.)</td>
</tr>
</tbody>
</table>
See Also

IFile Interface
SimpleEssentials.IO.Types Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
IFile Properties

The IFile type exposes the following members.
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Extension</td>
<td></td>
</tr>
<tr>
<td>Size</td>
<td></td>
</tr>
</tbody>
</table>
See Also

IFile Interface
SimpleEssentials.IO.Types Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
IFile extension property

Namespace: SimpleEssentials.IO.Types
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#
VB
C++

string Extension { get; set; }

Property Extension As String
    Get
    Set

property String^ Extension {
    String^ get ();
    void set (String^ value);
}
See Also

IFile Interface
SimpleEssentials.IO.Types Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
IFile..::..Size Property

Namespace:  SimpleEssentials.IO.Types
Assembly:  SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#

long Size { get; set; }

VB

Property Size As Long
    Get
    Set

C++

long Size { get; set; }

property long long^ Size {
    long long^ get ();
    void set (long long^ value);
}
See Also

IFile Interface
SimpleEssentials.IO.Types Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
IFileType Interface

**Namespace:** SimpleEssentials.IO.Types  
**Assembly:** SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#
VB
C++

public interface IFileType
Public Interface IFileType
public interface class IFileType
See Also

IFileType Members
SimpleEssentials.IO.Types Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
**IFileType Members**

The **IFileType** type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Inherited From</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Equals(System.Object)</strong></td>
<td>Determines whether the specified object is equal to the current object.</td>
<td><strong>Object</strong></td>
</tr>
<tr>
<td><strong>Exist</strong></td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection.</td>
<td><strong>Object</strong></td>
</tr>
<tr>
<td><strong>Finalize</strong></td>
<td>Serves as the default hash function.</td>
<td><strong>Object</strong></td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Sets the <strong>Type</strong> of the current instance.</td>
<td><strong>Object</strong></td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td>Creates a shallow copy of the current <strong>Object</strong>.</td>
<td><strong>Object</strong></td>
</tr>
<tr>
<td><strong>Load</strong></td>
<td>Returns a string that represents the current object.</td>
<td><strong>Object</strong></td>
</tr>
<tr>
<td><strong>MemberwiseClone</strong></td>
<td></td>
<td><strong>Object</strong></td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td></td>
<td><strong>Object</strong></td>
</tr>
</tbody>
</table>
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>FullPath</code></td>
<td></td>
</tr>
<tr>
<td><code>Loaded</code></td>
<td></td>
</tr>
<tr>
<td><code>Name</code></td>
<td></td>
</tr>
<tr>
<td><code>Type</code></td>
<td></td>
</tr>
</tbody>
</table>
See Also

IFileType Interface
SimpleEssentials.IO.Types Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
IFileType Methods

The IFileType type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Equals(System.Object)</code></td>
<td>Determines whether the specified object is equal to the current object. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><code>Exist</code></td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><code>Finalize</code></td>
<td>Serves as the default hash function. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><code>GetType</code></td>
<td>Gets the <code>Type</code> of the current instance. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><code>Load</code></td>
<td>Creates a shallow copy of the current <code>Object</code>.</td>
</tr>
<tr>
<td><code>MemberwiseClone</code></td>
<td>Returns a string that represents the current object. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><code>ToString</code></td>
<td></td>
</tr>
</tbody>
</table>
See Also

IFileType Interface
SimpleEssentials.IO.Types Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
IFileType...Exist Method

Namespace: SimpleEssentials.IO.Types
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#

```csharp
bool Exist(
    string path
)
```

Function Exist ( _
    path As String _
) As Boolean

```vbnet
bool^ Exist(
    String^ path
)
```

Parameters

path
    Type: String
See Also

**IFileType Interface**
**SimpleEssentials.IO.Types Namespace**

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
IFileType...Load Method

Namespace:  SimpleEssentials.IO.Types
Assembly:  SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#
VB
C++

bool Load(
    string path
)

Function Load ( _
    path As String _
) As Boolean

bool^ Load(
    String^ path
)

Parameters

path
    Type: String
See Also

IFileType Interface
SimpleEssentials.IO.Types Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
IFileType Properties

The IFileType type exposes the following members.
### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FullPath</td>
<td></td>
</tr>
<tr>
<td>Loaded</td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td></td>
</tr>
<tr>
<td>Type</td>
<td></td>
</tr>
</tbody>
</table>
See Also

IFileType Interface
SimpleEssentials.IO.Types Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
Namespace: SimpleEssentials.IO.Types
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#

VB

C++

string FullPath { get; set; }

Property FullPath As String
    Get
    Set

property String& FullPath {
    String& get ();
    void set (String& value);
}
See Also

IFileType Interface
SimpleEssentials.IO.Types Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
**IFileType..::..Loaded Property**

**Namespace:** SimpleEssentials.IO.Types  
**Assembly:** SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#
VB
C++

bool Loaded { get; set; }

Property Loaded As Boolean
  Get
  Set

property bool^ Loaded {
  bool^ get ();
  void set (bool^ value);
}

See Also

IFileType Interface
SimpleEssentials.IO.Types Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
IFileType...Name Property

Namespace: SimpleEssentials.IO.Types
Assembly: SimpleEssentials (in SimpleEssentials.dll)
**Syntax**

C#

```csharp
string Name { get; set; }
```

VB

```vbnet
Property Name As String
    Get
    Set
```

C++

```cpp
property String^ Name {
    String^ get ();
    void set (String^ value);
}
```
See Also

IFileType Interface
SimpleEssentials.IO.Types Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
IFileType....Type Property

Namespace:  SimpleEssentials.IO.Types
Assembly:  SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#
VB
C++

FileType Type { get; }

ReadOnly Property Type As FileType
Get

property FileType^ Type {
    FileType^ get ();
}
See Also

IFileType Interface
SimpleEssentials.IO.Types Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
IFolder Interface

Namespace: SimpleEssentials.IO.Types
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#
VB
C++

public interface IFolder : IFileType

Public Interface IFolder _
Inherits IFileType

public interface class IFolder : IFileType
See Also

SimpleEssentials.IO.Types Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
IFolder Methods

The IFolder type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Equals(System.Object)</code></td>
<td>Determines whether the specified object is equal to the current object. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><code>Finalize</code></td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><code>GetHashCode</code></td>
<td>Serves as the default hash function. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><code>GetType</code></td>
<td>Gets the <code>Type</code> of the current instance. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><code>MemberwiseClone</code></td>
<td>Creates a shallow copy of the current <code>Object</code>. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><code>ToString</code></td>
<td>Returns a string that represents the current object. (Inherited from <code>Object</code>.)</td>
</tr>
</tbody>
</table>
See Also

IFolder Interface
SimpleEssentials.IO.Types Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
SimpleEssentials.IO.Writers
Namespace
## Classes

<table>
<thead>
<tr>
<th>Class</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CsvWriter</td>
<td></td>
</tr>
</tbody>
</table>
## Interfaces

<table>
<thead>
<tr>
<th>Interface</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>~ IFileWriter</td>
<td></td>
</tr>
</tbody>
</table>

*Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.*
CsvWriter Class

Namespace: SimpleEssentials.IO.Writers
Assembly: SimpleEssentials (in SimpleEssentials.dll)
**Syntax**

### C#

```csharp
public class CsvWriter : IFileWriter
```

### VB

```vbnet
Public Class CsvWriter
    Implements IFileWriter
```

### C++

```cpp
public ref class CsvWriter : IFileWriter
```
Inheritance Hierarchy

Object
SimpleEssentials.IO.Writers...CsvWriter
See Also

CsvWriter Members
SimpleEssentials.IO.Writers Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
CsvWriter Members

The CsvWriter type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="System.Object" alt="" /> <strong>Equals</strong></td>
<td>Determines whether the specified object is equal to the current object.</td>
</tr>
<tr>
<td><img src="Object" alt="" /> <strong>Finalize</strong></td>
<td>Allows an object to try to free resources and perform other cleanup</td>
</tr>
<tr>
<td><img src="Object" alt="" /> <strong>GetHashCode</strong></td>
<td>Serves as the default hash function.</td>
</tr>
<tr>
<td><img src="Object" alt="" /> <strong>GetType</strong></td>
<td>Gets the <code>Type</code> of the current instance.</td>
</tr>
<tr>
<td><img src="Object" alt="" /> <strong>MemberwiseClone</strong></td>
<td>Creates a shallow copy of the current <code>Object</code>.</td>
</tr>
<tr>
<td><img src="Object" alt="" /> <strong>ToString</strong></td>
<td>Returns a string that represents the current object.</td>
</tr>
</tbody>
</table>

**Write** *(Of <<(T)>>)*

- *(String, T, Boolean)*
- *(String, IEnumerable<<(T)>>, Boolean)*
See Also

CsvWriter Class
SimpleEssentials.IO.Writers Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
CsvWriter Methods

The CsvWriter type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Equals(System.Object)</code></td>
<td>Determines whether the specified object is equal to the current object.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><code>Finalize</code></td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><code>GetHashCode</code></td>
<td>Serves as the default hash function.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><code>GetType</code></td>
<td>Gets the <code>Type</code> of the current instance.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><code>MemberwiseClone</code></td>
<td>Creates a shallow copy of the current <code>Object</code>.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><code>ToString</code></td>
<td>Returns a string that represents the current object.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><code>Write&lt;T&gt;</code></td>
<td>(Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><code>(String, T, Boolean)</code></td>
<td>(Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><code>Write&lt;T&gt;</code></td>
<td>(Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><code>(String, IEnumerable&lt;T&gt;)</code></td>
<td>(Inherited from <code>Object</code>.)</td>
</tr>
</tbody>
</table>

*Note:* The `Write` method allows writing to a stream and has overloaded versions for different types and arguments.
See Also

CsvWriter Class
SimpleEssentials.IO.Writers Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
CsvWriter...:::Write Method
# Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Write(Of &lt;&lt;'(T)&gt;&gt;)(String, T, Boolean)</code></td>
<td></td>
</tr>
</tbody>
</table>
| `Write(Of <<'(T)>>)(String, IEnumerable(Of <<'(T)>>), Boolean)` | }
See Also

CsvWriter Class
CsvWriter Members
SimpleEssentials.IO.Writers Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
CsvWriter...:::Write(Of (('T>))>
Method (String, T, Boolean)

Namespace: SimpleEssentials.IO.Writers
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#

```csharp
public void Write<T>(
    string filePath,
    T obj,
    bool append
)
```

VB

```vbnet
Public Sub Write(Of T) ( _
    filePath As String, _
    obj As T, _
    append As Boolean _
)
```

C++

```c++
public:
    template<typename T>
    void Write(
        String^ filePath,
        T obj,
        bool^ append
    )
```

**Type Parameters**

T

**Parameters**

filePath
   Type: String

obj
   Type: T

append
   Type: Boolean
See Also

CsvWriter Class
Write Overload
SimpleEssentials.IO.Writers Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
CsvWriter...:..Write(Of (Of ('T)>)> Method (String, IEnumerable(Of (Of ('T)>)>) Boolean)

Namespace: SimpleEssentials.IO.Writers
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#

```csharp
public void Write<T>(
    string filePath,
    IEnumerable<T> obj,
    bool append
)
```

VB

```vb
Public Sub Write(Of T) (  
    filePath As String,  
    obj As IEnumerable(Of T),  
    append As Boolean  
)
```

C++

```cpp
public:  
generic<typename T>  
void Write(  
    String^ filePath,  
    IEnumerable<T>^ obj,  
    bool^ append  
)
```

Type Parameters

T

Parameters

filePath
    Type: String

obj
    Type: IEnumerable<Of (Of T)>)

append
    Type: Boolean
See Also

CsvWriter Class
Write Overload
SimpleEssentials.IO.Writers Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
IFileWriter Interface

Namespace: SimpleEssentials.IO.Writers
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#

Public interface IFileWriter

VB

Public Interface IFileWriter

C++

public interface class IFileWriter
See Also

IFileWriter Members
SimpleEssentials.IO.Writers Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
The **IFileWriter** type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Equals(System.Object)</code></td>
<td>Determines whether the specified object is equal to the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td><code>Finalize</code></td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)</td>
</tr>
<tr>
<td><code>GetHashCode</code></td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td><code>GetType</code></td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td><code>MemberwiseClone</code></td>
<td>Creates a shallow copy of the current Object. Returns a string that represents the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td><code>ToString</code></td>
<td></td>
</tr>
<tr>
<td><code>Write(Of&lt;T&gt;&gt;()</code></td>
<td>(String, T, Boolean)</td>
</tr>
<tr>
<td><code>Write(Of&lt;'(T)')()</code></td>
<td>(String, IEnumerable(Of&lt;T&gt;), Boolean)</td>
</tr>
</tbody>
</table>
See Also

IFileWriter Interface
SimpleEssentials.IO.Writers Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
**IFileWriter Methods**

The IFileWriter type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Equals(System.Object)</strong></td>
<td>Determines whether the specified object is equal to the current object.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <strong>Object</strong>.)</td>
</tr>
<tr>
<td><strong>Finalize</strong></td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <strong>Object</strong>.)</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Serves as the default hash function.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <strong>Object</strong>.)</td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td>Gets the <strong>Type</strong> of the current instance.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <strong>Object</strong>.)</td>
</tr>
<tr>
<td><strong>MemberwiseClone</strong></td>
<td>Creates a shallow copy of the current <strong>Object</strong>.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <strong>Object</strong>.)</td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td>Returns a string that represents the current object.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <strong>Object</strong>.)</td>
</tr>
<tr>
<td><strong>Write&lt;(Of &lt;&lt;'(T)&gt;&gt;)</strong></td>
<td>(String, T, Boolean)</td>
</tr>
<tr>
<td><strong>Write&lt;(Of &lt;&lt;'(T)&gt;&gt;)</strong></td>
<td>(String, IEnumerable&lt;(Of &lt;&lt;'(T)&gt;&gt;), Boolean)</td>
</tr>
</tbody>
</table>
See Also

IFileWriter Interface
SimpleEssentials.IO.Writers Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
IFileWriter...Write Method
# Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Write&lt;(Of &lt;&lt;'(T)&gt;&gt;)(String, T, Boolean)</td>
<td></td>
</tr>
<tr>
<td>Write&lt;(Of &lt;&lt;'(T)&gt;&gt;)(String, IEnumerable&lt;(Of &lt;&lt;'(T)&gt;&gt;), Boolean)</td>
<td></td>
</tr>
</tbody>
</table>
See Also

IFileWriter Interface
IFileWriter Members
SimpleEssentials.IO.Writers Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
IFileWriter...:::Write<(Of <('<T'>)>))> Method (String, T, Boolean)

Namespace: SimpleEssentials.IO.Writers
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#

```csharp
void Write<T>(
    string filePath,
    T obj,
    bool append
)
```

VB

```vbnet
Sub Write(Of T) ( _
    filePath As String, _
    obj As T, _
    append As Boolean _
)
```

dummy

generic<typename T>

```csharp
void Write(<
    String^ filePath,
    T obj,
    bool^ append
)
```

Type Parameters

T

Parameters

filePath
  Type: String

obj
  Type: T

append
  Type: Boolean
See Also

IFileWriter Interface
Write Overload
SimpleEssentials.IO.Writers Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
IFileWriter...:::Write(Of <(<'T'>)>)
Method (String, IEnumerable(Of <(<'T'>)>), Boolean)

Namespace: SimpleEssentials.IO.Writers
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#
VB
C++

```csharp
void Write<T>(
    string filePath,
    IEnumerable<T> obj,
    bool append
)
```

```vbnet
Sub Write(Of T) (
    filePath As String,
    obj As IEnumerable(Of T),
    append As Boolean
)
```

generic<typename T>

```csharp
void Write(
    String^ filePath,
    IEnumerable<T>^ obj,
    bool^ append
)
```

**Type Parameters**

T

**Parameters**

`filePath`
Type: `String`

`obj`
Type: `IEnumerable<Of <(Of<T>)>>`

`append`
Type: `Boolean`
See Also

IFileWriter Interface
Write Overload
SimpleEssentials.IO.Writers Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
SimpleEssentials.LinqToSQL
Namespace
## Classes

<table>
<thead>
<tr>
<th>Class</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="#" alt="Generator" /></td>
<td>Generator</td>
</tr>
<tr>
<td><img src="#" alt="WherePart" /></td>
<td>WherePart</td>
</tr>
</tbody>
</table>

*Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.*
Generator Class

**Namespace:**  [SimpleEssentials.LinqToSQL](SimpleEssentials.LinqToSQL)
**Assembly:**  SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#

Public class Generator

VB

Public Class Generator

C++

public class Generator

public ref class Generator
Inheritance Hierarchy

Object
SimpleEssentials.LinqToSQL...Generator
See Also

Generator Members
SimpleEssentials.LinqToSQL Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
Generator Members

The Generator type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CreateTableSql&lt;Of &lt;&lt;(T)&gt;&gt;&gt;</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>Equals(System.Object)</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)</td>
</tr>
<tr>
<td>InsertAndReturnIdSql&lt;Of &lt;&lt;(T)&gt;&gt;&gt;</td>
<td></td>
</tr>
<tr>
<td>InsertSql&lt;Of &lt;&lt;(T)&gt;&gt;&gt;</td>
<td></td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>ToSql&lt;Of &lt;&lt;(T)&gt;&gt;&gt;</td>
<td>Returns a string that represents the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>(Expression&lt;Of &lt;&lt;(Func&lt;Of &lt;&lt;(T, Boolean)&gt;&gt;)&gt;&gt;)&gt;&gt;)</td>
<td></td>
</tr>
<tr>
<td>(Expression&lt;Of &lt;&lt;(Func&lt;Of &lt;&lt;(T, T2, Boolean)&gt;&gt;)&gt;&gt;)&gt;&gt;)</td>
<td></td>
</tr>
<tr>
<td>ToString</td>
<td></td>
</tr>
<tr>
<td>WhereSql&lt;Of &lt;&lt;(T)&gt;&gt;&gt;</td>
<td></td>
</tr>
</tbody>
</table>
See Also

Generator Class
SimpleEssentials.LinqToSQL Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
Generator Methods

The Generator type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CreateTableSql&lt;Of &lt;&lt;(T)&gt;&gt;)</td>
<td>Determines whether the specified object is equal to the current object.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from Object.)</td>
</tr>
<tr>
<td>Equals(System.Object)</td>
<td>Allows an object to try to free resources and perform other cleanup</td>
</tr>
<tr>
<td></td>
<td>operations before it is reclaimed by garbage collection.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from Object.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Serves as the default hash function.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from Object.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Gets the Type of the current instance.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from Object.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Creates a shallow copy of the current Object.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from Object.)</td>
</tr>
<tr>
<td>InsertAndReturnIdSql&lt;Of &lt;&lt;(T)&gt;&gt;)</td>
<td>Returns a string that represents the current object.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from Object.)</td>
</tr>
<tr>
<td>InsertSql&lt;Of &lt;&lt;(T)&gt;&gt;)</td>
<td></td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td></td>
</tr>
<tr>
<td>ToSql&lt;Of &lt;&lt;(T)&gt;&gt;)</td>
<td></td>
</tr>
<tr>
<td>(Expression&lt;Of &lt;&lt;(Func&lt;Of &lt;&lt;(T, Boolean)&gt;&gt;)&gt;&gt;)&gt;&gt;)</td>
<td></td>
</tr>
<tr>
<td>ToSql&lt;Of &lt;&lt;(T, T2)&gt;&gt;)</td>
<td></td>
</tr>
<tr>
<td>(Expression&lt;Of &lt;&lt;(Func&lt;Of &lt;&lt;(T, T2, Boolean)&gt;&gt;)&gt;&gt;)&gt;&gt;)</td>
<td></td>
</tr>
<tr>
<td>ToString</td>
<td></td>
</tr>
<tr>
<td>WhereSql&lt;Of &lt;&lt;(T)&gt;&gt;)</td>
<td></td>
</tr>
</tbody>
</table>
See Also

Generator Class
SimpleEssentials.LinqToSQL Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
Generator... CreateTableSql<Of<('T')>> Method

Namespace:  SimpleEssentials.LinqToSQL
Assembly:  SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#

public static string CreateTableSql<T>()

VB

Public Shared Function CreateTableSql(Of T) As String

C++

public:

generic<typename T>

static String^ CreateTableSql()
See Also

Generator Class
SimpleEssentials.LinqToSQL Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
Generator.....InsertAndReturnIdSql<
<('T')>> Method

Namespace: SimpleEssentials.LinqToSQL
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#  
**VB**  
C++

```csharp
public static string InsertAndReturnIdSql<T>(
    T obj
)
```

```vbnet
Public Shared Function InsertAndReturnIdSql(Of T) ( _
    obj As T _
) As String
```

```cpp
public:
    generic<typename T>
    static String^ InsertAndReturnIdSql(  
        T obj
    )
```

**Type Parameters**

T

**Parameters**

obj  
    Type: T
See Also

Generator Class
SimpleEssentials.LinqToSQL Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
Generator..::..InsertSql<Of<br><('T')>> Method

Namespace: SimpleEssentials.LinqToSQL
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#
VB
C++

public static string InsertSql<T>(
   T obj,
   Type overrideType
)

Public Shared Function InsertSql(Of T) ( _
   obj As T, _
   overrideType As Type _
) As String

public:
    generic<typename T>
    static String^ InsertSql(
       T obj,
       Type^ overrideType
    )

Type Parameters

T

Parameters

obj
   Type: T

overrideType
   Type: Type
See Also

Generator Class
SimpleEssentials.LinqToSQL Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
Generator::<@:ToSql(Of (<'T'>))>
Method (Expression(Of (<'Func(Of (<'T, Boolean'>)>>>)))

Namespace: SimpleEssentials.LinqToSQL
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#
public static WherePart ToSql<T>(
    Expression<Func<T, bool>> expression
)

VB
Public Shared Function ToSql(Of T)( _
    expression As Expression(Of Func(Of T, Boolean)) _
) As WherePart

C++
public:
  template<typename T>
  static WherePart^ ToSql(
    Expression<Func<T, bool>^>^> expression
  )

Type Parameters

T

Parameters

expression
  Type: Expression(Of (<'Func(Of (<'T, Boolean)>)>)>)}
See Also

Generator Class
ToSql Overload
SimpleEssentials.LinqToSQL Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
Generator...ToSql<(Of ("T, T2">
Method (Expression<(Of
<"Func<(Of ("T, T2,
Boolean">

Namespace: SimpleEssentials.LinqToSQL
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#
VB
C++

public static WherePart ToSql<T, T2>(
    Expression<Func<T, T2, bool>> expression
)

Public Shared Function ToSql(Of T, T2) ( _
    expression As Expression(Of Func(Of T, T2, Boolean)) _
) As WherePart

public:
  generic<typename T, typename T2>
  static WherePart^ ToSql(
    Expression<Func<T, T2, bool>^>^> expression
  )

Type Parameters

T
T2

Parameters

expression
    Type: Expression(Of (Of Func(Of (Of T, Boolean)>)>))>
See Also

Generator Class
ToSql Overload
SimpleEssentials.LinqToSQL Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
Generator...WhereSql<(<'T'>)> Method

Namespace:  SimpleEssentials.LinqToSQL
Assembly:  SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#  
VB  
C++

public static WherePart WhereSql<T>(
    Expression<Func<T, bool>> expression
)

Public Shared Function WhereSql(Of T) ( _
    expression As Expression(Of Func(Of T, Boolean)) _
) As WherePart

public:
    generic<typename T>
    static WherePart^ WhereSql(
        Expression<Func<T, bool>>^>^ expression
    )

Type Parameters

T

Parameters

expression
    Type: Expression<Of <'Func<Of <'T, Boolean>>>>)>
See Also

Generator Class
SimpleEssentials.LinqToSQL Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
WherePart Class

**Namespace:**  SimpleEssentials.LinqToSQL
**Assembly:**  SimpleEssentials (in SimpleEssentials.dll)
**Syntax**

C#  
Public class WherePart

VB  
Public Class WherePart

C++  
public ref class WherePart
Inheritance Hierarchy

Object
SimpleEssentials.LinqToSQL.......WherePart
See Also

WherePart Members
SimpleEssentials.LinqToSQL Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
WherePart Members

The WherePart type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Concat(String, WherePart)</strong></td>
<td>Determines whether the specified object is equal to the current object.</td>
</tr>
<tr>
<td><strong>Concat(WherePart, String, WherePart)</strong></td>
<td>(Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><strong>Equals(System.Object)</strong></td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><strong>Finalize</strong></td>
<td>Serves as the default hash function. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Gets the <code>Type</code> of the current instance. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td>Creates a shallow copy of the current <code>Object</code>. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><strong>IsCollection</strong></td>
<td>Returns a string that represents the current object. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><strong>IsParameter</strong></td>
<td></td>
</tr>
<tr>
<td><strong>IsSql</strong></td>
<td></td>
</tr>
<tr>
<td><strong>MemberwiseClone</strong></td>
<td></td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td></td>
</tr>
</tbody>
</table>
# Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Parameters" /></td>
<td>Parameters</td>
</tr>
<tr>
<td><img src="image" alt="Sql" /></td>
<td>Sql</td>
</tr>
</tbody>
</table>
See Also

WherePart Class
SimpleEssentials.LinqToSQL Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
WherePart Methods

The **WherePart** type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Concat(String, WherePart)</strong></td>
<td>Determines whether the specified object is equal to the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td><strong>Concat(WherePart, String, WherePart)</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Equals(System.Object)</strong></td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)</td>
</tr>
<tr>
<td><strong>Finalize</strong></td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td></td>
</tr>
<tr>
<td><strong>IsCollection</strong></td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td><strong>IsParameter</strong></td>
<td></td>
</tr>
<tr>
<td><strong>IsSql</strong></td>
<td></td>
</tr>
<tr>
<td><strong>MemberwiseClone</strong></td>
<td>Returns a string that represents the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td></td>
</tr>
</tbody>
</table>
See Also

WherePart Class
SimpleEssentials.LinqToSQL Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
WherePart:::..Concat Method
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>🔄 Concat(String, WherePart)</td>
<td></td>
</tr>
<tr>
<td>🔄 Concat(WherePart, String, WherePart)</td>
<td></td>
</tr>
</tbody>
</table>
See Also

WherePart Class
WherePart Members
SimpleEssentials.LinqToSQL Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
WherePart..::..Concat Method (String, WherePart)

Namespace:  SimpleEssentials.LinqToSQL
Assembly:  SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#
VB
C++

```csharp
public static WherePart Concat(
    string operator,
    WherePart operand
)
```

```vbnet
Public Shared Function Concat (
    operator As String,
    operand As WherePart
) As WherePart
```

```c++
public static WherePart^ Concat(
    String^ operator,
    WherePart^ operand
)
```

**Parameters**

operator
   Type: String

operand
   Type: SimpleEssentials.LinqToSQL..::.WherePart
See Also

WherePart Class
Concat Overload
SimpleEssentials.LinqToSQL Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
WherePart...:::Concat Method
(WherePart, String, WherePart)

Namespace: SimpleEssentials.LinqToSQL
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#
VB
C++

public static WherePart Concat(
    WherePart left,
    string operator,
    WherePart right
)

Public Shared Function Concat (_
    left As WherePart, _
    operator As String, _
    right As WherePart _
) As WherePart

public static WherePart^ Concat(
    WherePart^ left,
    String^ operator,
    WherePart^ right
)

Parameters

left
    Type: SimpleEssentials.LinqToSQL:::::WherePart

operator
    Type: String

right
    Type: SimpleEssentials.LinqToSQL:::::WherePart
See Also

WherePart Class
Concat Overload
SimpleEssentials.LinqToSQL Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
WherePart...::..IsCollection Method

Namespace: SimpleEssentials.LinqToSQL
Assembly: SimpleEssentials (in SimpleEssentials.dll)
### Syntax

**C#**
```
public static WherePart IsCollection(
    ref int countStart,
    IEnumerable values
)
```

**VB**
```
Public Shared Function IsCollection ( _
    ByRef countStart As Integer, _
    values As IEnumerable _
) As WherePart
```

**C++**
```
public:
static WherePart^ IsCollection(
    int^% countStart,
    IEnumerable^ values
)
```

### Parameters

**countStart**
- Type: `Int32`

**values**
- Type: `IEnumerable`
See Also

WherePart Class
SimpleEssentials.LinqToSQL Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
WherePart...:::IsParameter Method

Namespace: SimpleEssentials.LinqToSQL
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#

```csharp
public static WherePart IsParameter(int count,
                                    Object value)
```

VB

```vbnet
Public Shared Function IsParameter (_
    count As Integer, _
    value As Object _
) As WherePart
```

C++

```cpp
public:

static WherePart^ IsParameter(
    int^ count,
    Object^ value
)
```

Parameters

count
   Type: Int32

value
   Type: Object
See Also

**WherePart Class**
**SimpleEssentials.LinqToSQL Namespace**

*Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.*
WherePart...::..IsSql Method

**Namespace:**  [SimpleEssentials.LinqToSQL](https://www.simpleessentials.com.LinqToSQL)

**Assembly:**  SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#  
public static WherePart IsSql(       
    string sql       
)  

VB  
Public Shared Function IsSql ( _  
    sql As String _  
) As WherePart  

C++  
public:  
static WherePart^ IsSql(       
    String^ sql       
)  

Parameters

sql  
Type: String
See Also

WherePart Class
SimpleEssentials.LinqToSQL Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
WherePart Properties

The WherePart type exposes the following members.
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameters</td>
<td></td>
</tr>
<tr>
<td>Sql</td>
<td></td>
</tr>
</tbody>
</table>
See Also

*WherePart Class*

*SimpleEssentials.LinqToSQL Namespace*

*Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.*
WherePart...::..Parameters Property

Namespace: SimpleEssentials.LinqToSQL
Assembly: SimpleEssentials (in SimpleEssentials.dll)
### Syntax

**C#**
```csharp
public Dictionary<string, Object> Parameters { get; set; }
```

**VB**
```vbnet
Public Property Parameters As Dictionary(Of String, Object)
    Get
    Set
```

**C++**
```cpp
public:
    property Dictionary<String^, Object^>^ Parameters {
        Dictionary<String^, Object^>^ get ();
        void set (Dictionary<String^, Object^>^ value);
    }
```
See Also

WherePart Class
SimpleEssentials.LinqToSQL Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
WherePart...::...Sql Property

Namespace: SimpleEssentials.LinqToSQL
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#  
VB  
C++

public string Sql { get; set; }

Public Property Sql As String
Get
Set

public:
property String^ Sql {
    String^ get();
    void set (String^ value);
}

See Also

WherePart Class
SimpleEssentials.LinqToSQL Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
SimpleEssentials.Utils Namespace
## Classes

<table>
<thead>
<tr>
<th>Class</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ConsoleUtil</td>
<td></td>
</tr>
<tr>
<td>CustomCommand</td>
<td></td>
</tr>
<tr>
<td>PredicateBuilder</td>
<td></td>
</tr>
<tr>
<td>ProgressBar</td>
<td></td>
</tr>
<tr>
<td>Reflection</td>
<td></td>
</tr>
<tr>
<td>SqlBuilder</td>
<td></td>
</tr>
</tbody>
</table>

*Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.*
ConsoleUtil Class

Namespace:  SimpleEssentials_Utils
Assembly:  SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#

Public static class ConsoleUtil

VB

Public NotInheritable Class ConsoleUtil

C++

public ref class ConsoleUtil abstract sealed
Inheritance Hierarchy

Object
SimpleEssentials.Utils...ConsoleUtil
See Also

ConsoleUtil Members
SimpleEssentials_Utils Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
ConsoleUtil Members

The ConsoleUtil type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PadElementsInLines</td>
<td></td>
</tr>
</tbody>
</table>
See Also

ConsoleUtil Class
SimpleEssentials_Utils Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
ConsoleUtil Methods

The ConsoleUtil type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>🥁 <strong>PadElementsInLines</strong></td>
<td></td>
</tr>
</tbody>
</table>
See Also

ConsoleUtil Class
SimpleEssentials_Utils Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
ConsoleUtil::PadElementsInLines Method

Namespace: SimpleEssentials_Utis
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#
VB
C++

public static string PadElementsInLines(
    List<string[]> lines,
    int padding
)

Public Shared Function PadElementsInLines (_
    lines As List(Of String()), _
    padding As Integer _
) As String

public:
static String^ PadElementsInLines(
    List<array<String^>[][]) lines,
    int^ padding
)

Parameters

lines
    Type: List<(Of <( Of String[][])>)>

padding
    Type: Int32
See Also

ConsoleUtil Class
SimpleEssentials.Utils Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
CustomCommand Class

Namespace: SimpleEssentials.Utils
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#
VB
C++

public class CustomCommand
Public Class CustomCommand
public ref class CustomCommand
Inheritance Hierarchy

Object
SimpleEssentials.Utils...CustomCommand
See Also

CustomCommand Members
SimpleEssentials_Utils Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
CustomCommand Members

The CustomCommand type exposes the following members.
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Concat</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>Equals(System.Object)</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetCommand</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Returns a string that represents the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td></td>
</tr>
<tr>
<td>ToString</td>
<td></td>
</tr>
</tbody>
</table>
See Also

CustomCommand Class
SimpleEssentials_Utils Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
CustomCommand Methods

The CustomCommand type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Concat</td>
<td>Determines whether the specified object is equal to the current object.</td>
</tr>
<tr>
<td>Equals(System.Object)</td>
<td>(Inherited from Object.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup</td>
</tr>
<tr>
<td></td>
<td>operations before it is reclaimed by garbage collection.</td>
</tr>
<tr>
<td>GetCommand</td>
<td>Serves as the default hash function.</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>(Inherited from Object.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from Object.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from Object.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from Object.)</td>
</tr>
</tbody>
</table>
See Also

CustomCommand Class
SimpleEssentials.Utils Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
CustomCommand...:..Concat Method

Namespace:  SimpleEssentials.Utils
Assembly:  SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#

```csharp
public void Concat(
    string command
)
```

VB

```vbnet
Public Sub Concat (_
    command As String _
)
```

C++

```cpp
public:
void Concat(
    String^ command
)
```

**Parameters**

command
    Type: String
See Also

CustomCommand Class
SimpleEssentials_Utils Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
CustomCommand::..GetCommand Method

Namespace: SimpleEssentials.Utils
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#
VB
C++

public string GetCommand()

Public Function GetCommand As String

public:
String^ GetCommand()
See Also

CustomCommand Class
SimpleEssentials.Utils Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
PredicateBuilder Class

Namespace: SimpleEssentials_Utils
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#
VB
C++

public static class PredicateBuilder
Public NotInheritable Class PredicateBuilder
public ref class PredicateBuilder abstract sealed
Inheritance Hierarchy

Object
SimpleEssentials.Utils.....PredicateBuilder
See Also

PredicateBuilder Members
SimpleEssentials_Utils Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
PredicateBuilder Members

The `PredicateBuilder` type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>And&lt;(Of &lt;&lt;'(T)&gt;&gt;)&gt;</td>
<td></td>
</tr>
<tr>
<td>False&lt;(Of &lt;&lt;'(T)&gt;&gt;)</td>
<td></td>
</tr>
<tr>
<td>Or&lt;(Of &lt;&lt;'(T)&gt;&gt;)&gt;</td>
<td></td>
</tr>
<tr>
<td>True&lt;(Of &lt;&lt;'(T)&gt;&gt;)&gt;</td>
<td></td>
</tr>
</tbody>
</table>
See Also

PredicateBuilder Class
SimpleEssentials_Utils Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
**PredicateBuilder Methods**

The **PredicateBuilder** type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>And&lt;(Of &lt;&lt;'(T)&gt;&gt;)&gt;&gt;</code></td>
<td></td>
</tr>
<tr>
<td><code>False&lt;(Of &lt;&lt;'(T)&gt;&gt;)&gt;&gt;</code></td>
<td></td>
</tr>
<tr>
<td><code>Or&lt;(Of &lt;&lt;'(T)&gt;&gt;)&gt;&gt;</code></td>
<td></td>
</tr>
<tr>
<td><code>True&lt;(Of &lt;&lt;'(T)&gt;&gt;)&gt;&gt;</code></td>
<td></td>
</tr>
</tbody>
</table>
See Also

PredicateBuilder Class
SimpleEssentials.Utils Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
PredicateBuilder...:::And<(Of
<('T'>))> Method

Namespace:  SimpleEssentials_Utils
Assembly:  SimpleEssentials (in SimpleEssentials.dll)
## Syntax

### C#
```csharp
public static Expression<Func<T, bool>> And<T>(
    Expression<Func<T, bool>> expr1,
    Expression<Func<T, bool>> expr2
)
```

### VB
```vbnet
Public Shared Function And(Of T) ( _
    expr1 As Expression(Of Func(Of T, Boolean)), _
    expr2 As Expression(Of Func(Of T, Boolean)) _
) As Expression(Of Func(Of T, Boolean))
```

### C++
```cpp
public:
    generic<typename T>
    static Expression<Func<T, bool>>^ And(
        Expression<Func<T, bool>>^ expr1,
        Expression<Func<T, bool>>^ expr2
    )
```

## Type Parameters

**T**

## Parameters

- **expr1**
  
  Type: Expression<(Of <('Func<(Of <('T, Boolean)>)>)>)>

- **expr2**
  
  Type: Expression<(Of <('Func<(Of <('T, Boolean)>)>)>)>
See Also

PredicateBuilder Class
SimpleEssentials_Utils Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
PredicateBuilder...:..False<(Of<br>\(<'T'>\)>)> Method

Namespace:  SimpleEssentials.Utils
Assembly:  SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#

```csharp
public static Expression<Func<T, bool>> False<T>()
```

VB

```vbnet
Public Shared Function False(Of T) As Expression(Of Func(Of T, Boolean)
```

C++

```cpp
public:
  generic<typename T>
  static Expression<Func<T, bool>>^False()
```

Type Parameters

T
See Also

PredicateBuilder Class  
SimpleEssentials_Utils Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
PredicateBuilder...:::Or<(Of
<('T')>)> Method

Namespace: SimpleEssentials.Utils
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#

```
public static Expression<Func<T, bool>> Or<T>(
    Expression<Func<T, bool>> expr1,
    Expression<Func<T, bool>> expr2
)
```

VB

```
Public Shared Function Or(Of T) ( _
    expr1 As Expression(Of Func(Of T, Boolean)), _
    expr2 As Expression(Of Func(Of T, Boolean)) _
) As Expression(Of Func(Of T, Boolean))
```

C++

```
public:
    template<typename T>
    static Expression<Func<T, bool>> Or(
        Expression<Func<T, bool>> expr1,
        Expression<Func<T, bool>> expr2
    )
```

Type Parameters

T

Parameters

expr1
    Type: Expression<Of <'(Func<Of <'(T, Boolean)>)>>>

expr2
    Type: Expression<Of <'(Func<Of <'(T, Boolean)>)>>>
See Also

PredicateBuilder Class
SimpleEssentials.Utils Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
PredicateBuilder...:::True(Of (<'T'>)> Method

Namespace:  SimpleEssentials_Utils
Assembly:  SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#  

public static Expression<Func<T, bool>> True<T>()

VB  

Public Shared Function True(Of T) As Expression(Of Func(Of T, Boolean)

c++

public:

generic<typename T>

static Expression<Func<T, bool>>^ True();

Type Parameters

T
See Also

PredicateBuilder Class
SimpleEssentials_Utils Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
ProgressBar Class

Namespace: SimpleEssentials_Utils
Assembly: SimpleEssentials (in SimpleEssentials.dll)
## Syntax

### C#

```csharp
public class ProgressBar : IDisposable
```

### VB

```vbnet
Public Class ProgressBar
    Implements IDisposable
```

### C++

```cpp
public ref class ProgressBar : IDisposable
```
Inheritance Hierarchy

Object
SimpleEssentials.Utils:::ProgressBar
See Also

ProgressBar Members
SimpleEssentials_Utils Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
ProgressBar Members

The ProgressBar type exposes the following members.
## Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ProgressBar()()()</td>
<td></td>
</tr>
<tr>
<td>ProgressBar(String)</td>
<td></td>
</tr>
</tbody>
</table>
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dispose</td>
<td>Determines whether the specified object is equal to the current object.</td>
</tr>
<tr>
<td>Equals(System.Object)</td>
<td>(Inherited from Object.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection.</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function.</td>
</tr>
<tr>
<td>GetType</td>
<td>(Inherited from Object.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object.</td>
</tr>
<tr>
<td>Report(Double)</td>
<td>(Inherited from Object.)</td>
</tr>
<tr>
<td>Report(Double, String)</td>
<td></td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from Object.)</td>
</tr>
</tbody>
</table>
See Also

ProgressBar Class
SimpleEssentials_Utils Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
ProgressBar Constructor
# Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ProgressBar</td>
<td></td>
</tr>
<tr>
<td>ProgressBar(String)</td>
<td></td>
</tr>
</tbody>
</table>
See Also

ProgressBar Class
ProgressBar Members
SimpleEssentials.Utils Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
ProgressBar Constructor

Namespace: SimpleEssentials_Utils
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

**C#**

```csharp
public ProgressBar()
```

**VB**

```vbnet
Public Sub New
```

**C++**

```cpp
public:
ProgressBar()
```
See Also

ProgressBar Class
ProgressBar Overload
SimpleEssentials_Utils Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
ProgressBar Constructor (String)

**Namespace:** SimpleEssentials_Utils

**Assembly:** SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#
VB
C++

public ProgressBar(
   string label
)

Public Sub New (_
   label As String _
)

public:
   ProgressBar(
   String^ label
)

Parameters

label
   Type: String
See Also

ProgressBar Class
ProgressBar Overload
SimpleEssentials.Utils Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
ProgressBar Methods

The ProgressBar type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dispose</td>
<td>Determines whether the specified object is equal to the current object.</td>
</tr>
<tr>
<td>Equals(System.Object)</td>
<td>(Inherited from Object.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection.</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function.</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance.</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object.</td>
</tr>
<tr>
<td>Report(Double)</td>
<td>(Inherited from Object.)</td>
</tr>
<tr>
<td>Report(Double, String)</td>
<td>(Inherited from Object.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from Object.)</td>
</tr>
</tbody>
</table>
See Also

ProgressBar Class
SimpleEssentials_Utils Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
ProgressBar... Dispose Method

Namespace: SimpleEssentials_Utils
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#
VB
C++

public void Dispose()

Public Sub Dispose

public:
void Dispose()
See Also

ProgressBar Class
SimpleEssentials.Utils Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
ProgressBar:::Report Method
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Report(Double)</td>
<td></td>
</tr>
<tr>
<td>Report(Double, String)</td>
<td></td>
</tr>
</tbody>
</table>
See Also

ProgressBar Class
ProgressBar Members
SimpleEssentials_Utils Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
ProgressBar...Report Method (Double)

Namespace: SimpleEssentials_Utils
Assembly: SimpleEssentials (in SimpleEssentials.dll)
**Syntax**

**C#**

```csharp
public void Report(
    double value
)
```

**VB**

```vbnet
Public Sub Report (_
    value As Double _
)
```

**C++**

```c++
public:
void Report(
    double^ value
)
```

**Parameters**

value  
Type: **Double**
See Also

ProgressBar Class
Report Overload
SimpleEssentials_Utils Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
ProgressBar...Report Method (Double, String)

Namespace: SimpleEssentials.Utils
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#
VB
C++

class
	public void Report(
		double value,
		string label
	)
	
Public Sub Report ( _
		value As Double, _
		label As String _
	)

class
	public:

t: void Report(
		double^ value,
		String^ label
	)

Parameters

value
	Type: Double

label
	Type: String
See Also

ProgressBar Class
Report Overload
SimpleEssentials.Utils Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
Reflection Class

Namespace: SimpleEssentials.Utils
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

**C#**

`public static class Reflection`

**VB**

`Public NotInheritable Class Reflection`

**C++**

`public ref class Reflection abstract sealed`
Inheritance Hierarchy

Object
SimpleEssentials.Utils...Reflection
See Also

Reflection Members
SimpleEssentials_Utils Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
Reflection Members

The Reflection type exposes the following members.
# Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GenerateCreateColumns</td>
<td></td>
</tr>
<tr>
<td>GenerateInsertColumnNames</td>
<td></td>
</tr>
<tr>
<td>GetIdentityProperty</td>
<td></td>
</tr>
<tr>
<td>GetKeyName</td>
<td></td>
</tr>
<tr>
<td>GetNonIdentityProperties</td>
<td></td>
</tr>
<tr>
<td>GetTableName</td>
<td></td>
</tr>
<tr>
<td>IsIdentityMember</td>
<td></td>
</tr>
<tr>
<td>IsNullable</td>
<td></td>
</tr>
<tr>
<td>TypeToSqlType</td>
<td></td>
</tr>
</tbody>
</table>
See Also

Reflection Class
SimpleEssentials_Utils Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
Reflection Methods

The Reflection type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GenerateCreateColumns</td>
<td></td>
</tr>
<tr>
<td>GenerateInsertColumnNames</td>
<td></td>
</tr>
<tr>
<td>GetIdentityProperty</td>
<td></td>
</tr>
<tr>
<td>GetKeyName</td>
<td></td>
</tr>
<tr>
<td>GetNonIdentityProperties</td>
<td></td>
</tr>
<tr>
<td>GetTableName</td>
<td></td>
</tr>
<tr>
<td>IsIdentityMember</td>
<td></td>
</tr>
<tr>
<td>IsNullable</td>
<td></td>
</tr>
<tr>
<td>TypeToSqlType</td>
<td></td>
</tr>
</tbody>
</table>
See Also

Reflection Class
SimpleEssentials_Utils Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
Reflection..::..GenerateCreateColumn Method

Namespace: SimpleEssentials.Utils
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#

public static string GenerateCreateColumns(
    Object obj,
    Type overrideType
)

VB

Public Shared Function GenerateCreateColumns ( _
    obj As Object, _
    overrideType As Type _
) As String

C++

public:

static String^ GenerateCreateColumns(
    Object^ obj,
    Type^ overrideType
)

Parameters

obj

    Type: Object

overrideType

    Type: Type
See Also

Reflection Class
SimpleEssentials_Utils Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
Reflection..::.GenerateInsertColumn
Method

Namespace: SimpleEssentials_Utils
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

**C#**

public static string GenerateInsertColumnNames(
    Object obj,
    Type overrideType
)

**VB**

Public Shared Function GenerateInsertColumnNames ( _
    obj As Object, _
    overrideType As Type _
) As String

**C++**

public:
static String^ GenerateInsertColumnNames(
    Object^ obj,
    Type^ overrideType
)

**Parameters**

- **obj**
  - Type: `Object`

- **overrideType**
  - Type: `Type`
See Also

Reflection Class
SimpleEssentials_Utils Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
Reflection..::..GetIdentityProperty Method

Namespace: SimpleEssentials_Utils
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#  
public static Member GetIdentityProperty(
    Object obj,
    Type overrideType
)

VB  
Public Shared Function GetIdentityProperty (_
    obj As Object, _
    overrideType As Type _
) As Member

C++  
public:  
static Member^ GetIdentityProperty(
    Object^ obj,
    Type^ overrideType
)

Parameters

obj  
Type: Object

overrideType  
Type: Type
See Also

Reflection Class
SimpleEssentials.Utils Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
Reflection..::..GetKeyName Method

Namespace:  SimpleEssentials_Utils
Assembly:  SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#

```csharp
public static string GetKeyName(
    Object obj,
    string propNameOverride
)
```

VB

```vbnet
Public Shared Function GetKeyName ( _
    obj As Object, _
    propNameOverride As String _
) As String
```

C++

```cpp
public:
static String^ GetKeyName(
    Object^ obj,
    String^ propNameOverride
)
```

Parameters

obj
  Type: `Object`

propNameOverride
  Type: `String`
See Also

Reflection Class
SimpleEssentials.Utils Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
Reflection:::GetNonIdentityProperties Method

Namespace: SimpleEssentials.Utils
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#  
VB  
C++

public static List<Member> GetNonIdentityProperties(
    Object obj,
    Type overrideType
)

Public Shared Function GetNonIdentityProperties ( _
    obj As Object, _
    overrideType As Type _
) As List(Of Member)

public:
static List<Member>^ GetNonIdentityProperties(
    Object^ obj,
    Type^ overrideType
)

Parameters

obj
    Type: Object

overrideType
    Type: Type
See Also

Reflection Class
SimpleEssentials.Utils Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
Reflection...::..GetTableName Method

Namespace: SimpleEssentials_Utils
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#

public static string GetTableName(
    Object obj,
    Type overrideType
)

VB

Public Shared Function GetTableName ( _
    obj As Object, _
    overrideType As Type _
) As String

C++

public:

static String^ GetTableName(
    Object^ obj,
    Type^ overrideType
)

Parameters

obj
    Type: Object

overrideType
    Type: Type
See Also

Reflection Class
SimpleEssentials.Utils Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
Reflection::..IsIdentityMember Method

Namespace: SimpleEssentials_Utils
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#

Public Shared Function IsIdentityMember ( _
    member As Member _
) As Boolean

VB

Public Shared Function IsIdentityMember ( _
    member As Member _
) As Boolean

C++

public static bool IsIdentityMember(
    Member member
)

Parameters

member
    Type: Member
See Also

Reflection Class
SimpleEssentials.Utils Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
Reflection..::.IsNullable Method

Namespace:  SimpleEssentials_Utils
Assembly:  SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#
VB
C++

public static bool IsNullable(
    Type type
)

Public Shared Function IsNullable ( _
    type As Type _
) As Boolean

public:
static bool^ IsNullable(
    Type^ type
)

Parameters

type
    Type: Type
See Also

Reflection Class
SimpleEssentials_Utils Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
Reflection..::.::..TypeToSqlType Method

Namespace: SimpleEssentials_Utils
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#

VB

C++

public static string TypeToSqlType(
    Type type
)

Public Shared Function TypeToSqlType ( _
    type As Type _
) As String

public:
static String^ TypeToSqlType(
    Type^ type
)

Parameters

type
    Type: Type
See Also

Reflection Class
SimpleEssentials.Utils Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
SqlBuilder Class

Namespace:  SimpleEssentials.Utils
Assembly:  SimpleEssentials (in SimpleEssentials.dll)
Syntax

**C#**

```csharp
public static class SqlBuilder
```

**VB**

```vbnet
Public NotInheritable Class SqlBuilder
```

**C++**

```cpp
public ref class SqlBuilder abstract sealed
```
Inheritance Hierarchy

Object
  SimpleEssentials.Utils...SqlBuilder
See Also

SqlBuilder Members
SimpleEssentials_Utils Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
SqlBuilder Members

The SqlBuilder type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>$ Generate(WherePart)</code></td>
<td></td>
</tr>
<tr>
<td><code>$ Generate(CustomCommand)</code></td>
<td></td>
</tr>
<tr>
<td><code>$ Join&lt;(Of &lt;&lt;(T, T2)&gt;&gt;)</code></td>
<td></td>
</tr>
<tr>
<td><code>$ JoinLeft&lt;(Of &lt;&lt;(T, T2)&gt;&gt;)</code></td>
<td></td>
</tr>
<tr>
<td><code>$ On&lt;(Of &lt;&lt;(T, T2)&gt;&gt;)</code></td>
<td></td>
</tr>
<tr>
<td><code>$ Select&lt;(Of &lt;&lt;(T)&gt;&gt;)</code></td>
<td></td>
</tr>
<tr>
<td><code>$ Where&lt;(Of &lt;&lt;(T)&gt;&gt;)</code></td>
<td></td>
</tr>
</tbody>
</table>
See Also

SqlBuilder Class
SimpleEssentials.Utils Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
SqlBuilder Methods

The SqlBuilder type exposes the following members.
# Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Generate(WherePart)</td>
<td></td>
</tr>
<tr>
<td>Generate(CustomCommand)</td>
<td></td>
</tr>
<tr>
<td>Join(Of &lt;&lt;(T, T2)&gt;&gt;)</td>
<td></td>
</tr>
<tr>
<td>JoinLeft(Of &lt;&lt;(T, T2)&gt;&gt;)</td>
<td></td>
</tr>
<tr>
<td>On(Of &lt;&lt;(T, T2)&gt;&gt;)</td>
<td></td>
</tr>
<tr>
<td>Select(Of &lt;&lt;(T)&gt;&gt;)</td>
<td></td>
</tr>
<tr>
<td>Where(Of &lt;&lt;(T)&gt;&gt;)</td>
<td></td>
</tr>
</tbody>
</table>
See Also

SqlBuilder Class
SimpleEssentials_Utils Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
SqlBuilder...:..Generate Method
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Generate(WherePart)</td>
<td></td>
</tr>
<tr>
<td>Generate(CustomCommand)</td>
<td></td>
</tr>
</tbody>
</table>
See Also

SqlBuilder Class
SqlBuilder Members
SimpleEssentials.Utils Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
SqlBuilder...:..Generate Method (WherePart)

Namespace: SimpleEssentials_Utils
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#

public static WherePart Generate(
    WherePart wherePart
)

VB

Public Shared Function Generate ( _
    wherePart As WherePart _
) As WherePart

C++

public:
static WherePart^ Generate(
    WherePart^ wherePart
)

Parameters

wherePart

Type: SimpleEssentials.LinqToSQL::WherePart
See Also

- SqlBuilder Class
- Generate Overload
- SimpleEssentials.Utils Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
SqlBuilder...:..Generate Method (CustomCommand)

Namespace: SimpleEssentials_Utils
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#

VB

C++

```csharp
public static WherePart Generate(CustomCommand command)
```

```vbnet
Public Shared Function Generate(command As CustomCommand)
    As WherePart
```

```c++
public: static WherePart Generate(CustomCommand command)
```

**Parameters**

command

Type: SimpleEssentials.Utils::CustomCommand
See Also

SqlBuilder Class
Generate Overload
SimpleEssentials.Utils Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
SqlBuilder:::..Join(Of <(<'T, T2'>)> Method

Namespace:  SimpleEssentials_Utils
Assembly:  SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#

```csharp
public static CustomCommand Join<T, T2>(
    CustomCommand command, 
    Expression<Func<T, T2, bool>> expression
)
```

VB

```vbnet
Public Shared Function Join(Of T, T2) ( _
    command As CustomCommand, _
    expression As Expression(Of Func(Of T, T2, Boolean)) _
) As CustomCommand
```

C++

```cpp
public:

    explicit static 
    CustomCommand^ Join(
        
        CustomCommand^ command, 
        Expression<Func<T, T2, bool>^>^ expression
    
    )
```

Type Parameters

T
T2

Parameters

command

Type: `SimpleEssentials.Utils...::CustomCommand`

eexpression

Type: `Expression<Of (Of Func<Of (Of 'T, T2, Boolean)>)>)>`
See Also

SqlBuilder Class  
SimpleEssentials.Utils Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
SqlBuilder::JoinLeft(Of (<'T, T2>))> Method

Namespace: SimpleEssentials_Utils
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#
VB
C++

public static CustomCommand JoinLeft<T, T2>(
    CustomCommand command,
    Expression<Func<T, T2, bool>> expression
)

Public Shared Function JoinLeft(Of T, T2) ( _
    command As CustomCommand, _
    expression As Expression(Of Func(Of T, T2, Boolean)) _) _
) As CustomCommand

public:
    generic<typename T, typename T2>
    static CustomCommand^ JoinLeft(
        CustomCommand^ command,
        Expression<Func<T, T2, bool>^>^ expression
    )

Type Parameters

T
T2

Parameters

command
    Type: SimpleEssentials.Utils....CustomCommand

expression
    Type: Expression<(Of <'Func<(Of <('T, T2, Boolean)>)>>>)
See Also

SqlBuilder Class
SimpleEssentials.Utils Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
SqlBuilder...:...On<(Of <("T, T2")] Method

Namespace:  SimpleEssentials_Utils
Assembly:  SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#  
VB  
C++

public static CustomCommand On<T, T2>(
    CustomCommand command,
    Expression<Func<T, T2, bool>> expression
)

Public Shared Function On(Of T, T2) ( _
    command As CustomCommand, _
    expression As Expression(Of Func(Of T, T2, Boolean)) _
) As CustomCommand

public:
    generic<typename T, typename T2>
    static CustomCommand^ On(
        CustomCommand^ command,
        Expression<Func<T, T2, bool^>^ expression
    )

Type Parameters

T
T2

Parameters

command
    Type: SimpleEssentials_Utils..::..CustomCommand

eexpression
    Type: Expression<(Of <'Func<(Of <(<'T, T2, Boolean>>)>)>)>
See Also

SqlBuilder Class  
SimpleEssentials.Utils Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
SqlBuilder...::..Select(Of (T))> Method

Namespace:  SimpleEssentials_Utils
Assembly:  SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#
VB
C++

public static CustomCommand Select<T>()

Public Shared Function Select(Of T) As CustomCommand

public:
generic<typename T>
static CustomCommand^ Select()

Type Parameters

T
See Also

SqlBuilder Class
SimpleEssentials_Utils Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.
SqlBuilder...: Where(Of (Of 'T)>)

Method

Namespace: SimpleEssentials.Utils
Assembly: SimpleEssentials (in SimpleEssentials.dll)
Syntax

C#

```csharp
public static WherePart Where<T>(
    CustomCommand command,
    Expression<Func<T, bool>> expression
)
```

VB

```vb
Public Shared Function Where(Of T) (_
    command As CustomCommand, _
    expression As Expression(Of Func(Of T, Boolean)) _
) As WherePart
```

C++

```cpp
public:

generic<typename T>

static WherePart^ Where(
    CustomCommand^ command,
    Expression<Func<T, bool>^>^ expression
)
```

Type Parameters

T

Parameters

command

Type: SimpleEssentials.Utils..::.CustomCommand

expression

Type: Expression(Of ((Func(Of ((T, Boolean))))))
See Also

SqlBuilder Class
SimpleEssentials_Utils Namespace

Created with evaluation copy of GhostDoc Pro. Click here to purchase and remove this link.